Computation Theory

نظرية الحساب

تأليف

د.مصباح جمعة عقل   د.محمد خليل أبو زلطة

د.زياد عبد الكريم القاضي

الطبعة الأولى

2009م – 1430هـ



مكتبة المجتمع العربي للنشر والتوزيع

الفهرس

المحتوى | **الصفحة**

الوحدة الأولى

مدخل الى نظرية الحساب

الوحدة الثانية

آلة الحالة المنتهية

الوحدة الثالثة

آلة الحالة المنتهية غير المحدودة

المحتوى **الصفحة**

الوحدة الرابعة

آلة الحالة المنتهية المستخدمة للحزمة

الوحدة الخامسة

آلة يتورينج

الوحدة السادسة

آلة مور وآلة ميلي

المقدمة

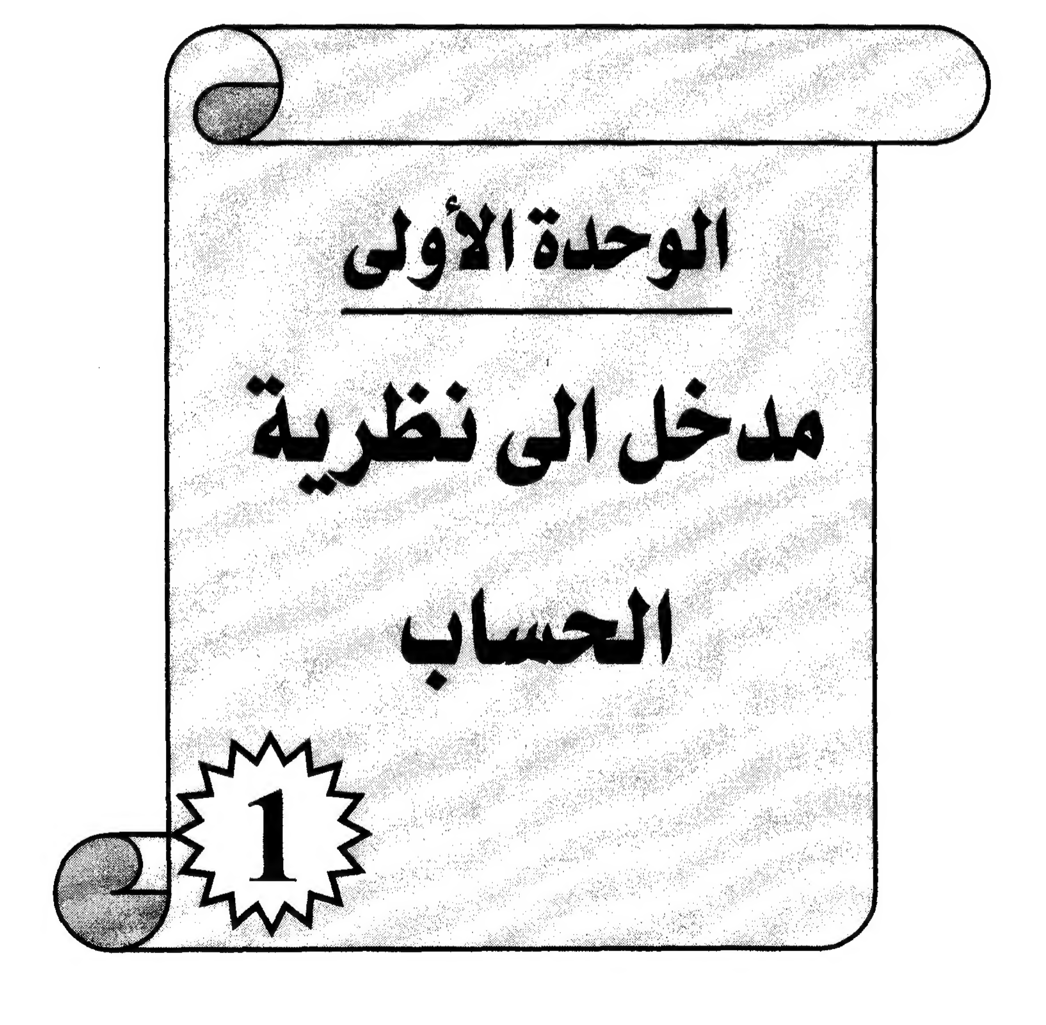
تستخدم مفاهيم نظرية الحساب في كثير من التطبيقات العملية حيث تستخدم في تصميم الدارات المنطقية وفي تصميم المترجمات ومعالجات النصوص والانظمة البرمجية التي تعتمد على عملية تمييز الانماط مثل انظمة معالجة الصور الرقمية وانظمة معالجة الصوت وانظمة الذكاء الصناعي المختلفة.

ونظرا لاهمية نظرية الحساب واستخداماتها المختلفة في تطبيقات علم الحاسوب وهندسته وغيره من تطبيقات فق جاء هذا الكتاب لتعريف القارئ العربي باهمية هذه النظرية واطلاعه على اهم مواضيعها ونخص بالذكر:

- الات الحالة المنتهية المحدودة.
- الات الحالة المنتهية غيرالمحدودة.
- الة الحالة باستخدام الحزمة.
- الة تيورينج.
- الة مورو الة ميلي.

هذا وقد حاولنا الاكثار من الامثلة التوضيحية املا منا في تسهيل فهم مبادئ نظرية الحساب املين نكون قد وفقنا في ايصال المعلومة بشكل واضح وسهل.

والله ولي التوفيق......


الوحدة الأولى
مدخل الى نظرية
الحساب
1

تستخدم في نظرية الحساب بعض الواضيع الرياضية ومن هذه المواضيع:

1 اساسيات المنطق.
2 المجموعات.
3 الاقترانات.
4 العلاقات.
5 التعابير المنتظمة.

وسوف نستعرض في هذه الوحدة و بشكل مختصر هذه المواضيع نظرا لاستخداماتها الكثيرة في الوحدات اللاحقة علما بانه يمكن الرجوع الى كتاب الرياضيات المنفصلة للمؤلفين وذلك مزيدا للمعلومات.

1.1 المنطق الاقتراحي Propositional Logic:

يعرف الاقتراح Proposition على انه جملة تصريحية تاخذ قيمة الصواب اوالخطا فمثلا الجمل التالية صحيحة:

6 العدد 10 هو عدد زوجي.
7 العدد 4 هو مربع كامل.
8 عمان عاصمة الاردن.
9 في حين تكون الجمل التالية خاطئة:
10 جرش عاصمة الاردن.
11 العدد 3 عدد زوجي.
12 يقبل العدد 20 القسمة عل 9 بدون باقي.

هذا ويمكن ربط الجل التصريحية معا باستخدام مجموعة من العلاقات المنطقية مثل علاقة "و" and وتاخذ الجملة التصريحية هنا ايضا قيمة واحدة هي الصواب او الخطا فمثلا الجمل التالية صحيحة:

- عمان او جرش مدينة اردنية.
- العدد 10 زوجي ويقبل القسمة على 5 بدون باقي.

اما الجمل التالية فهي خاطئة:

- عمان وجرش عاصمة الاردن.
- العدد 21 فردي و قبل القسمة على 4 بدون باقي.

وفيما يلي سوف نستعرض اهم العمليات المنطقية المستخدمة لربط الجمل:

1. علاقة الضرب المنطقي **Conjunction(And)**

تستخدم هذه العلاقة لربط اكثر من جملة وتكون النتيجة النهائية للجملة صحيحة اذا كانت كافة الجمل المرتبطة بهذه العلاقة صحيحة ويبين جدول الصواب التالي عملية الربط باستخدام هذه العلاقة:

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

2. علاقة الجمع المنطقي Disjunction (OR)

تستخدم هذه العلاقة لربط اكثر من جملة و تكون النتيجة صحيحة اذا كانت على الاقل قيمة احدى الجمل صحيحة ويبين الجدول التالي جدول الصواب لهذه العلاقة:

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

3. علاقة النفي Negation

وتستخدم هذه العلاقة لنفي جملة لتصبح النتيجة صحيحة اذا كانت الجملة خاطئة وبالعكس ويبين الجدول التالي جدول الصواب لهذه العلاقة:

p	$\neg p$
T	F
F	T

4. العلاقة الشرطية احادية الاتجاه Conditional

وتربط هذه العلاقة بين جملتين وتكون النتيجة صائبة اذا كانت الجملة الثانية صحيحة اوكانت الجملتان خاطئتين والجدول التالي يبين جدول الصواب لهذه العلاقة:

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

5. العلاقة الشرطية ثنائية الاتجاه Biconditional

تربط هذه العلاقة بين جملتين وتكون النتيجة صائبة اذا تشابهت قيم الجملتين ويبين الجدول التالي جدول الصواب لهذه العلاقة:

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

6. الحقيقة Tautology

جملة اواكثر مرتبطة معا وتكون نتيجتها دائما صحية مثل:

$p \vee \neg p$

7. التناقض Contradiction

جملة او اكثر مرتبطة معا وتكون نتيجتها دائما خاطئة مثل:

$p \wedge \neg p$

وفيما يلي اهم قواعد المنطق والممثلة للحقائق:

List of Identities:

1. $P \Leftrightarrow (P \vee P)$ ----- idempotence of $\vee$
2. $P \Leftrightarrow (P \wedge P)$ ----- idempotence of $\wedge$
3. $(P \vee Q) \Leftrightarrow (Q \vee P)$ ----- commutativity of $\vee$
4. $(P \wedge Q) \Leftrightarrow (Q \wedge P)$ ----- commutativity of $\wedge$
5. $[(P \vee Q) \vee R] \Leftrightarrow [P \vee (Q \vee R)]$ ----- associativity of $\vee$
6. $[(P \wedge Q) \wedge R] \Leftrightarrow [P \wedge (Q \wedge R)]$ ----- associativity of $\wedge$
7. $\neg(P \vee Q) \Leftrightarrow (\neg P \wedge \neg Q)$ ----- DeMorgan's Law
8. $\neg(P \wedge Q) \Leftrightarrow (\neg P \vee \neg Q)$ ----- DeMorgan's Law
9. $[P \wedge (Q \vee R] \Leftrightarrow [(P \wedge Q) \vee (P \wedge R)]$ ----- distributivity of $\wedge$ over $\vee$
10. $[P \vee (Q \wedge R] \Leftrightarrow [(P \vee Q) \wedge (P \vee R)]$ ----- distributivity of $\vee$ over $\wedge$
11. $(P \vee True) \Leftrightarrow True$
12. $(P \wedge False) \Leftrightarrow False$
13. $(P \vee False) \Leftrightarrow P$
14. $(P \wedge True) \Leftrightarrow P$
15. $(P \vee \neg P) \Leftrightarrow True$
16. $(P \wedge \neg P) \Leftrightarrow False$
17. $P \Leftrightarrow \neg(\neg P)$ ----- double negation
18. $(P \rightarrow Q) \Leftrightarrow (\neg P \vee Q)$ ----- implication
19. $(P \leftrightarrow Q) \Leftrightarrow [(P \rightarrow Q) \wedge (Q \rightarrow P)]$----- equivalence
20. $[(P \wedge Q) \rightarrow R] \Leftrightarrow [P \rightarrow (Q \rightarrow R)]$ ----- exportation
21. $[(P \rightarrow Q) \wedge (P \rightarrow \neg Q)] \Leftrightarrow \neg P$ ----- absurdity
22. $(P \rightarrow Q) \Leftrightarrow (\neg Q \rightarrow \neg P)$ ----- contrapositive

اما قوانين الاستنتاج الاساسية فهي كما يلي:

List of Implications:

1. $P \Rightarrow (P \vee Q)$ ----- addition
2. $(P \wedge Q) \Rightarrow P$ ----- simplification

3. $[P \wedge (P \rightarrow Q)] \Rightarrow Q$ ----- modus ponens
4. $[(P \rightarrow Q) \wedge \neg Q] \Rightarrow \neg P$ ----- modus tollens
5. $[\neg P \wedge (P \vee Q)] \Rightarrow Q$ ----- disjunctive syllogism
6. $[(P \rightarrow Q) \wedge (Q \rightarrow R)] \Rightarrow (P \rightarrow R)$ ----- hypothetical syllogism
7. $(P \rightarrow Q) \Rightarrow [(Q \rightarrow R) \rightarrow (P \rightarrow R)]$
8. $[(P \rightarrow Q) \wedge (R \rightarrow S)] \Rightarrow [(P \wedge R) \rightarrow (Q \wedge S)]$
9. $[(P \leftrightarrow Q) \wedge (Q \leftrightarrow R)] \Rightarrow (P \leftrightarrow R)$

وفيما يلي بعض الامثلة التوضيحية:

1. Construct the truth tables for the following tables for the following statements, and use the results to find logical implicationsand logical equivalences among them(say which statements imply which others, and which are equivalent to which others).

a. $p \rightarrow q) \wedge (p \rightarrow \neg q)$
b. $p \vee (p \rightarrow q)$
c. $p \wedge (p \rightarrow q)$
d. $p \rightarrow q) \wedge (\bar{\ }p \rightarrow q)$
e. $p \leftrightarrow (p \leftrightarrow q)$
f. $q \wedge (p \rightarrow q)$

Solution:

Here are the solutions for a, c & e...

(a)$(p \rightarrow q) \wedge (p \rightarrow \neg q)$

Assuming you have gone through the notes material explaining the basic concepts

Of Logic, we start building the truth tables without much explanation of as to how the truth values came.

p	q	q¬	q→p	q¬→p
T	T	F	T	F
T	F	T	F	T
F	T	F	T	T
F	F	T	T	T

q) →(c) p ^(p

p	q	q→p	q)→p ^(p
T	T	T	T
T	F	F	F
F	T	T	F
F	F	T	F

↔(p↔p (e) q)

p	q	q↔p	↔(p ↔p q)
T	T	T	T
T	F	F	F
F	T	F	T
F	F	T	F

1.14) Show that the statements p V q V r V s and(¬p ^ ¬q ^ ¬r) → s are equivalent.

Solution:

p	q	r	s	p V q V r V s
T	T	T	T	T
T	T	T	F	T
T	T	F	T	T
T	T	F	F	T
T	F	T	T	T
T	F	T	F	T
T	F	F	T	T
T	F	F	F	T
F	T	T	T	T
F	T	T	F	T
F	T	F	T	T
F	T	F	F	T
F	F	T	T	T
F	F	T	F	T
F	F	F	T	T
F	F	F	F	F

p	q	R	s	p¬	q¬	r¬	s→r ¬q^¬p^¬
T	T	T	T	F	F	F	T
T	T	T	F	F	F	F	T
T	T	F	T	F	F	T	T
T	T	F	F	F	T	T	T

T	F	T	T	F	T	F	T
T	F	T	F	F	T	F	T
T	F	F	T	F	T	T	T
T	F	F	F	F	T	T	T
F	T	T	T	T	F	F	T
F	T	T	F	T	F	F	T
F	T	F	T	T	F	T	T
F	T	F	F	T	F	T	T
F	F	T	T	T	T	F	T
F	F	T	F	T	T	T	T
F	F	F	T	T	T	T	T
F	F	F	F	T	T	T	F

The truth values for $p \vee q \vee r \vee s$ &$(\neg p \wedge \neg q \wedge \neg r) \rightarrow s$ are same in each case, then we can conclude that $p \vee q \vee r \vee s$ and$(\neg p \wedge \neg q \wedge \neg r) \rightarrow s$ are logically equivalent, written as

$p \vee q \vee r \vee s \Leftrightarrow (\neg p \wedge \neg q \wedge \neg r) \rightarrow s$

2.1المجموعات Sets:

تعـرف المجموعـة علـى انهـا مجموعـة مـن العناصـر ويمكـن ان تكـون هـذه العناصر من نفس النوع او يمكن ان تكون مختلفة.

وفيما يلي بعض الامثلة على المجموعات:

1. The set of students in this class
2. The set N of natural number(all non-negative integers) {0, 1, 2, 3, …}

3. The set Z of all integers both positive and negative {..., -2, -1, 0, 1, 2, ...}
4. The set Q of all rational numbers(numbers that can be expressed as p/q, where p and q are elements of Z
5. The set R of real numbers.
6. The set C of complex numbers

وفيما يلي اهم الصيغ الرياضية المرتبطة بالمجموعات:

1. اذا كان العنصر من بين عناصر المجموعة فان هذا العنصر ينتمي اله هذه المجموعة.
2. تكون المجموعة خالية اذا لم تحتوي على اي عنصر.
3. تكون المجموعة منتهية اذا كان عدد العناصر محدد.
4. تكون المجموعة لانهائية اذا مان عدد عناصرها غير منتهي.
5. ترتيب العناصر في المجموعة غير مهم.
6. اذا تكررت القيمة الواحدة كعناصر في المجموعة فانها تشكل عنصرا واحدا ويمكن الغاء التكرار دون التاثير على المجموعة.
7. يطلق على عدد العناصر في المجموعة العمق او cardinality.
8. اذا كان احد عناصر المجموعة عنصرا فان هذه المجموعة الجزئية تعتبر عنصرا واحدا وتنتمي للمجموعة الاصلية.

وفيما يلي بعض الامثلة والتي توضح هذه المفاهيم:

$A=\{1,3,5,7,9\}$ $\quad 1 \in A, 1 \in B, 1 \in C$
$B=\{x|x \text{ is odd}\}$
$C=\{1,3,5,7,9,\dots\}$
cardinality of $A=5$ $(|A|=5)$
A is a proper subset of B. $\quad A \subset B$
C is a subset of B. $\quad C \subseteq B$

9. اذا كانت مجموعة منتمية لمجموعة اخرى فان اي عنصر فيها ينتمي الى المجموعة الاخرى.

10. تكون المجموعتان متساويتين اذا احتوت كل منهما على نفس العناصر:

Sets and Subsets

subsets $A \subseteq B \Leftrightarrow \forall x[x \in A \Rightarrow x \in B]$

$A \not\subset B \Leftrightarrow \neg\forall x[x \in A \Rightarrow x \in B]$

$\Leftrightarrow \exists x \neg[\neg(x \in A) \vee x \in B)]$

$\Leftrightarrow \exists x[x \in A \wedge x \notin B]$

set equality $C = D \Leftrightarrow (C \subseteq D) \wedge (D \subseteq C)$

$C \neq D \Leftrightarrow \neg(C \subseteq D \wedge D \subseteq C)$

$\Leftrightarrow C \not\subseteq D \vee D \not\subseteq C$

المجموعة الاسية Power set

المجموعة الاسية لمجموعة ما هي مجموعة عدد عناصرها مساو 2 مرفوعا لاس مساو عدد عناصر المصفوفة:

If $|A|=n$, then $|P(A)|=2^n$.

مثال:

if $X = \{a, b, c\}$ then

$P(X) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$

الضرب الديكارتي لمجموعتين:

لضرب الديكارتي لمجموعتين هو مجموعة عناصرها تشكل كافة الاحتمالات الممكنة لتوليف عناصر المجموعتين.

مثال

if A = {1, 2, 3} and B = {a, b}, then A x B = {(1, a),(1, b),(2, a),(2, b),(3, a),(3, b)}.

امثلة

Example

$A = \{a, e, i, o, u\}$ is a set and the list of all its elements is given.

Example

$B = \{x : x \text{ is an integer}, x > 0\}$

Consider the set $C = \{1, 3, 5, 7, 9\}$. We write

$3 \in C$ to mean that 3 belongs to the set C, and

$-5 \notin C$ to mean that -5 does not belong to C.

Example

Let $A = \{a, e, i, o, u\}$, $B = \{u, o, i, e, a\}$ and $C = \{a, a, e, i, i, o, u\}$

then $A = B = C$

Example

Let $X = \{ y : \quad y^2 = 4 \ , \ y \text{ is odd} \}$

then X is the empty set and we write

$X = \varnothing$.

Example

Let $A = \{1, 3, 5\}$ and $B = \{1, 2, 3, 4, 5\}$

then $A \subset B$ and $B \not\subset A$

Example

Let $A = \{1, 3, 5, 7\}$ and $B = \{2, 4, 6, 7\}$

then A and B are not disjoint because 7 is in both sets:

$7 \in A$ and $7 \in B$

وفيما يلي اهم المجموعات الشائعة الاستخدام:

a. Z=the set of integers= {0, 1,-1, 2,-1, 3,-3...}
b. N=the set of nonnegative integers or natural numbers
c. Z+=the set of positive integers
d. Q=the set of rational numbers= {a/b| a,b is integer, b not zero}
e. Q+=the set of positive rational numbers
f. Q*=the set of nonzero rational numbers
g. R=the set of real numbers
h. R+=the set of positive real numbers
i. R*=the set of nonzero real numbers
j. C=the set of complex numbers

اسئلة:

Q1: $U = N. \{x \mid \forall y(y \geq x)\} =?$

Q2: $U = Z. \{x \mid \forall y(y \geq x)\} =?$

Q3: $U = Z. \{x \mid \exists y(y \in R \wedge y\,2 = x)\} =?$

Q4: $U = Z. \{x \mid \exists y\,(y \in R \wedge y\,3 = x)\} =?$

Q5: $U = R. \{|x| \mid x \in Z\} =?$

Q6: $U = R. \{\,|x|\,\} = ?$

A1: $U = N. \{X \mid \forall y\,(y \geq x)\} = \{0\}$

A2: $U = Z. \{x \mid \forall y\,(y \geq x)\} = \{\ \}$

A3: $U = Z. \{x \mid \exists y\,(y \in R \wedge y\,2 = x)\}$

$= \{0, 1, 2, 3, 4, \ldots\} = N$

A4: $U = Z. \{x \mid \exists y\,(y \in R \wedge y\,3 = x)\} = Z$

A5: $U = R. \{|x| \mid x \in Z\} = N$

A6: $U = R. \{|x|\}$ = non-negative reals.

العمليات الاساسية على المجموعات:

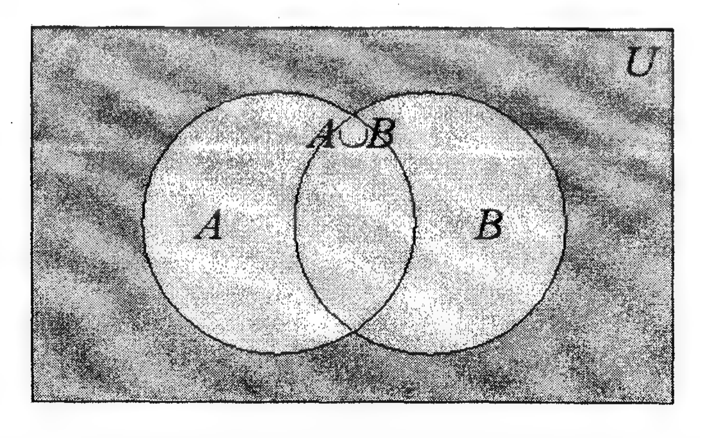
تنفذ على المجموعات العمليات الاساسية التالية:

امثلة:

1. الاتحاد
2. التقاطع
3. الفرق
4. النفي
5. الفرق المتماثل او عملية الاستبعاد

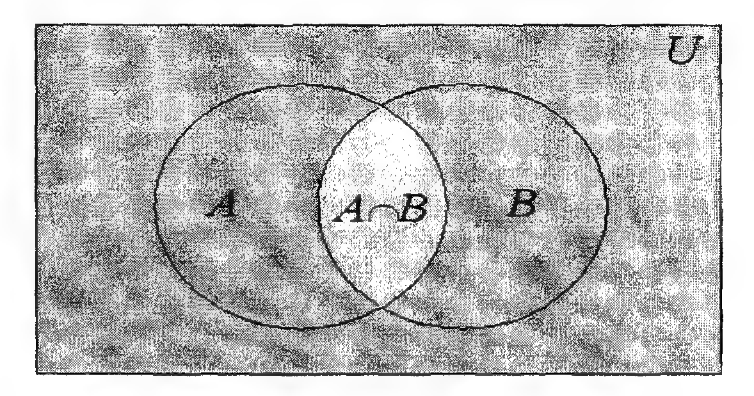
بمثل اتحاد مجموعتين مجموعة عناصرها هي عناصر المجموعة الاولى مضافا اليها غناصر المجموعة الثانية والتي لا تقع في المجموعة الاولى ويمكن تمثيل هذه العملية بمخططات فين كما يلي:

$$A \cup B = \{ x \mid x \in A \vee x \in B \}$$



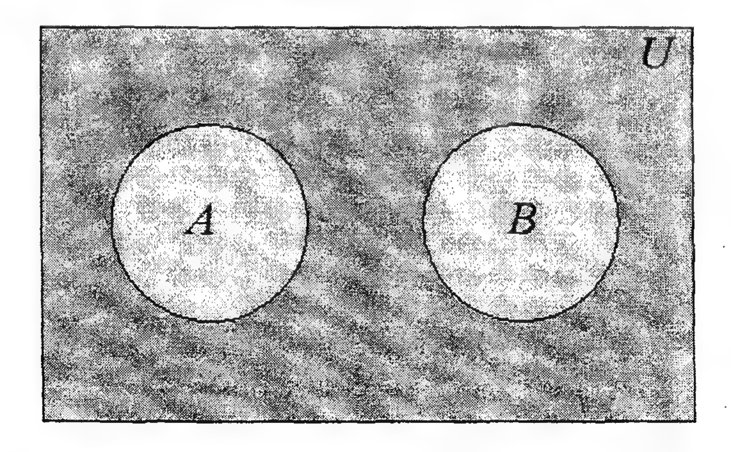
تقاطع مجموعتين هو مجموعة عناصرها هي العناصر التي تقع في المجموعة الاولى وفي نفس الوقت تقع في المجموعة الثانية وفيما يلي كيفية تمثيل هذه العملية:

$$A \cap B = \{ x \mid x \in A \wedge x \in B \}$$



اذا لم تحتوي المجموعتان على عناصر مشتركة فان تقاطعهما عبارة عن مجموعة خالية:

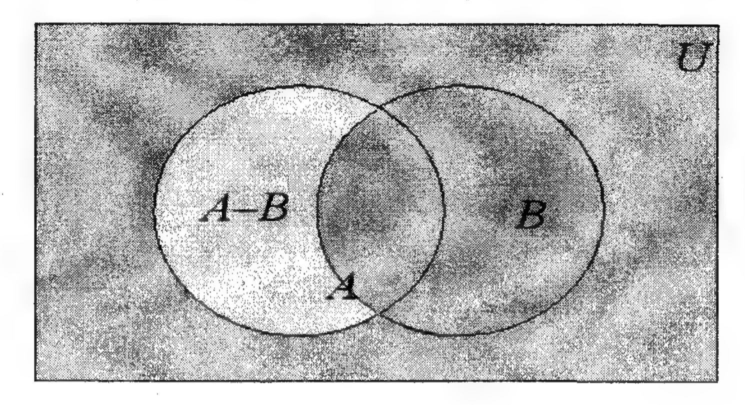
$$A \cap B = \emptyset .$$



الفرق:

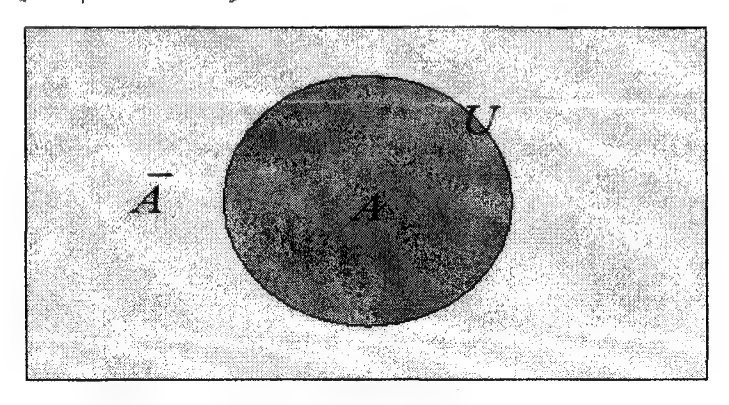
اذا طرحت مجموعة اولى من مجموعة ثانية فان الناتج مجموعة عناصرها هي عناصر المجموعة الثانية والتي لا تقع في المجموعة الاولى:

$$A-B = \{ x \mid x \in A \wedge x \notin B \}$$



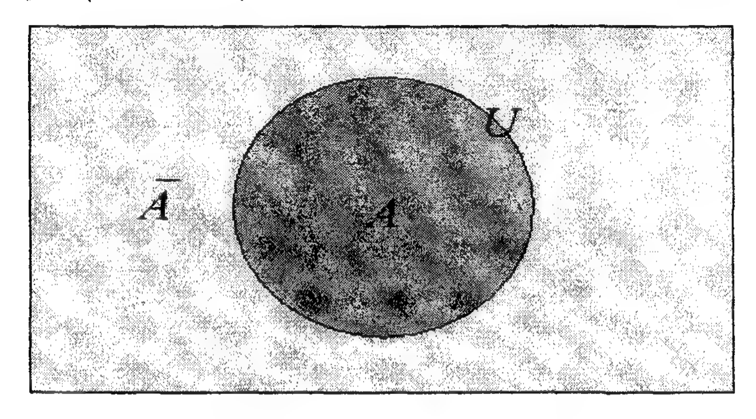
نفي المجموعة هو مجموعة عبارة عن عناصر المجموعة الكاملة(العالمية) مطروحا منه عناصر المجموعة:

$$\bar{A} = \{ x \mid x \notin A \}$$



الطرح المتماثل يعطي مجموعة عبارة عن حاصل جمع عناصر المجموعة الاولى والثانية باستثناء العناصر المشتركة:

$$\bar{A} = \{ x \mid x \notin A \}$$



امثلة:

Example

Let $A = \{ a, b, c \}$, $B = \{ 1, 2, 3 \}$ and $C = \{ a, c, e, 1, 3, 5 \}$.

then $A \cup B = \{ a, b, c, 1, 2, 3 \}$

$B \cup C = \{ 1, 2, 3, a, c, e, 5 \}$

$C \cup A = \{ a, c, e, 1, 3, 5, b \}$

Example

Let $A = \{a, b, c, d, e\}$, $B = \{c, d, e, f, g\}$ and $C = \{a, e, i, o, u\}$.

then $A \cap B = \{c, d, e\}$

$B \cap C = \{e\}$

$C \cap A = \{a, e\}$

Example

Let $S = \{a, b, c, d\}$ and $T = \{c, d, e, f\}$,

then $S \setminus T = \{a, b\}$

$T \setminus S = \{e, f\}$

Example

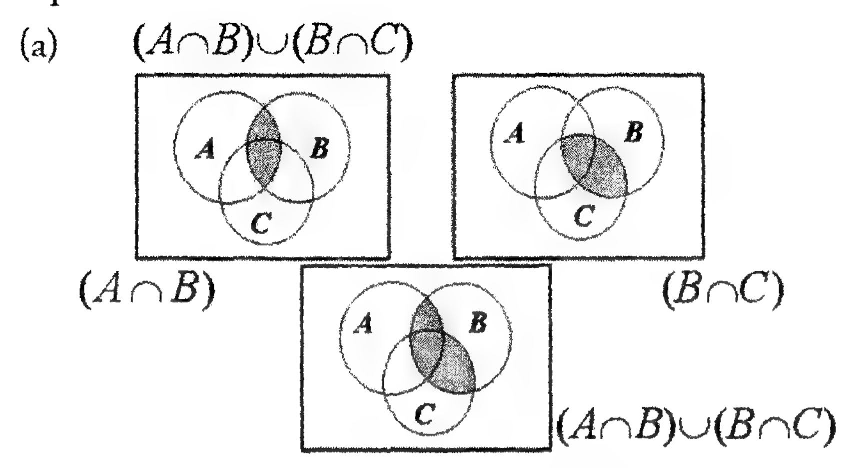
Let the universal set U be the set containing letters of the English alphabet and $A = \{a, b, c, x, y, z\}$.

then $\overline{A} = \{d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w\}$

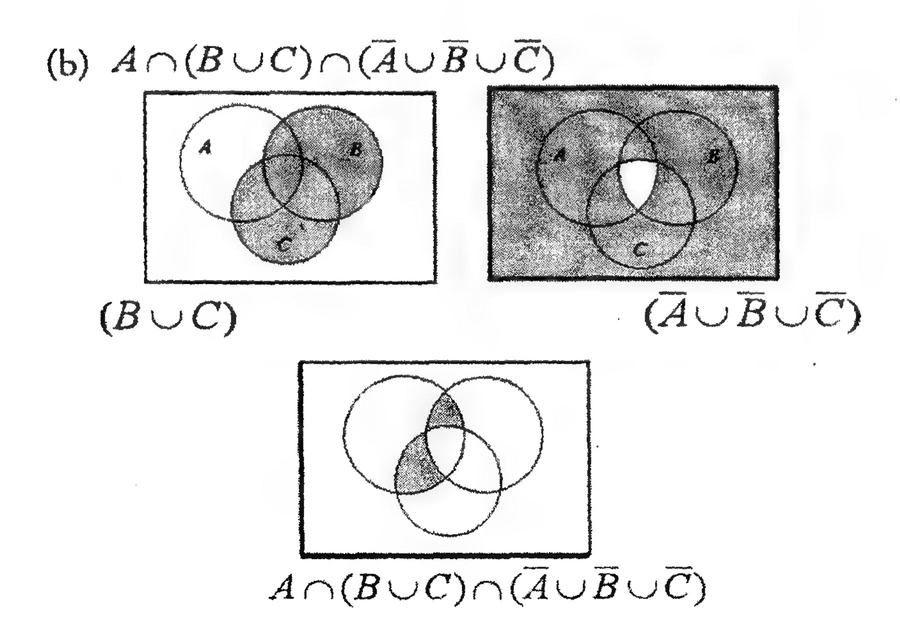
Example

Use Venn diagrams to represent the following set expressions.

(a) $(A \cap B) \cup (B \cap C)$



(b) $A \cap (B \cup C) \cap (\overline{A} \cup \overline{B} \cup \overline{C})$



قوانين المجموعات

تستخدم مجموعة من القاونين لتنقيد العمليات المختلفة على المجموعات وفيما يلي اهم هذه القوانين:

(1) $\overline{\overline{A}} = A$ Law of *Double Complement*

(2) $\overline{A \cup B} = \overline{A} \cap \overline{B}$ *Demorgan's* Laws

$\overline{A \cap B} = \overline{A} \cup \overline{B}$

(3) $A \cup B = B \cup A$ *Commutative* Laws

$A \cap B = B \cap A$

(4) $A \cup (B \cup C) = (A \cup B) \cup C$ *Associative* Laws

$A \cap (B \cap C) = (A \cap B) \cap C$

(5) $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ *Distributive* Laws

$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

(6) $A \cup A = A, A \cap A = A$ *Idempotent Laws*

(7) $A \cup \phi = A, A \cap U = A$ *Identity Laws*

(8) $A \cup \overline{A} = U, A \cap \overline{A} = \phi$ *Inverse Laws*

(9) $A \cup U = U, A \cap \phi = \phi$ *Domination Laws*

(10) $A \cup (A \cap B) = A$ *Absorption Laws*

$A \cap (A \cup B) = A$

امثلة:

Example

Let A, B and C be sets. Use laws of algebra of sets to simplify the following set expressions.

(a) $(A \cap B) \cup (B \cap C)$

$= (A \cap B) \cup (C \cap B)$ Commutative law

$= (A \cup C) \cap B$ Distributive law

(b) $(A \cap B) \cup (A \cap B \cap \overline{C} \cap D) \cup (\overline{A} \cap B)$

$= (A \cap B) \cup (\overline{A} \cap B) \cup (A \cap B \cap \overline{C} \cap D)$ Commutative law

$= ((A \cup \overline{A}) \cap B)) \cup (A \cap B \cap \overline{C} \cap D)$ Distributive law

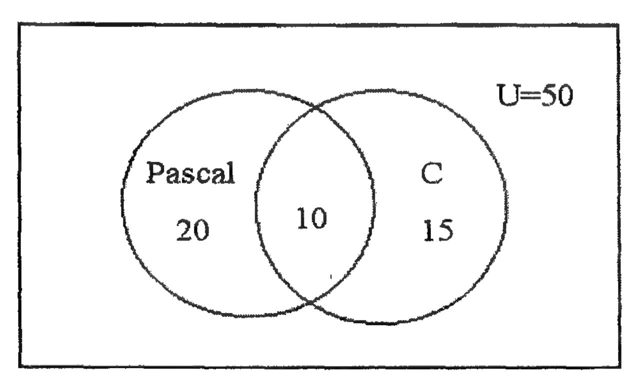
$= (U \cap B) \cup (A \cap B \cap \overline{C} \cap D)$ Inverse law

$= B \cup (A \cap B \cap \overline{C} \cap D)$ Identity law

$= B$ Absorption law

Example

In a class of 50 college students, 30 study Pascal, 25 study C and 10 study both computer languages. How many students do not study computer language ?



No. of students who do not study computer language is

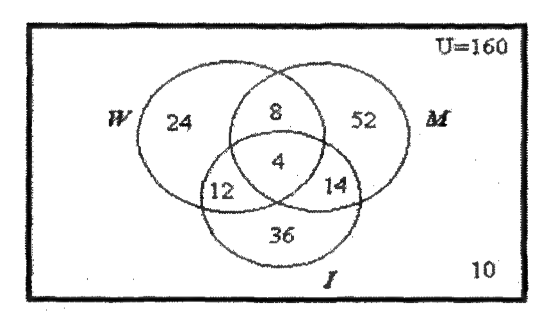
$50 - 20 - 10 - 15 = 5$ students

Example

In a survey of 160 passengers, an airline found that 48 preferred wine with their meals, 78 preferred mixed drinks, and 66 preferred ice tea. In addition, 12 enjoyed wine and mixed drinks, 18 enjoyed mixed drinks and ice tea, and 16 enjoyed ice tea and wine, and 4 passengers enjoyed them all.

a) How many passengers want only iced tea with their meals?

b) How many passengers do not like any of them?



a) No. of passengers = 36

b) No. of passengers

= 160 – 24 – 52 – 36 – 12 – 8 – 14 – 4

= 10

Problem:

Without using the Venn Diagrams, show that the symmetric difference operation $\oplus$ satisfies the Associative Property.You may use basic properties of set operations (union, intersections,complementing) such as commutativity, associativity, distributivity and De Morgan's laws without proof.

Solution:

We need to prove for arbitrary sets A, B and C $(A \oplus B) \oplus C = A \oplus (B \oplus C)$.

Note that $A \oplus B = (A \cap \overline{B}) \cup (\overline{A} \cap B)$ and $\overline{(A \oplus B)} = (A \cap B) \cup (\overline{A} \cap \overline{B})$

Hence the left hand side = $((A \oplus B) \cap \overline{C}) \cup \overline{(A \oplus B)} \cap C$

$= (((A \cap \overline{B}) \cup (\overline{A} \cap B)) \cap \overline{C}) \cup (((A \cap B) \cup (\overline{A} \cap \overline{B})) \cap C)$

$= (A \cap B \cap C) \cup (\overline{A} \cap \overline{B} \cap C) \cup (A \cap \overline{B} \cap \overline{C}) \cup (\overline{A} \cap B \cap \overline{C})$

Similarly it can be shown that the right hand side is equal to this last expression. Thus the Associativity holds.

3.1العلاقات Relations:

لتكن كل A,B مجموعة عندها فان العلاقة التي تربط هاتين المجموعتين هي حاصل الضرب الكارتيزي لهاتين المجموعتين.

مثال

لناخذ المجموعة التالية:

A = {2, 3, 5, 6)

العلاقة التي تربط عناصر هذه المجموعة بحيث يقسم العدد الثاني على الاول بدون باقي هي:

$R = \{(2, 2),(3, 3),(5, 5),(6, 6),(2,6),(3, 6)\}$.

هذه ويمكن الرجوع لكتاب الرياضيات المنفصلة لمزيد من المعلومات عن الاعلاقات والاقترانات وما يهمن هنا هو القاء نظرة سريعة على اهم خصائص العلاقات الا وهي:

الانعكاس

R is reflexive if for every $a \in A$, a R a.

التماثل

R is symmetric if for every a and b in A, if aRb, then bRa.

التعدي

R is transitive if for every a, b and c in A, if aRb and bRc, then aRc.

التساوي او التكافؤ

R is an equivalence relation on A if R is reflexive, symmetric and transitive

وفيما يلي بعض الامثلة التوضيحية:

1. In each case, a relation on the set {1, 2, 3} is given. Of the three properties, reflexivity, symmetry, and transitivity, determine which ones the relation has.

Give reasons for each of them.

a. R = {(1, 3), (3, 1), (2, 2)}.
b. R = {(1, 1), (2, 2), (3, 3), (1, 2)}.

Solution:

a. It is Symmetry .Because wherever there is something like(a, b)we also have(b, a)Here it is(1, 3) we also have(3, 1) and(2, 2).
b. It is reflexive ,because for every 'a' we have(a, a).Here it is(1, 1),(2,2),(3, 3).

2. Three relations are given on the set of all non-empty subsets of N. In each case, say whether the relation is Reflexive or Symmetric or it is Transitive.

a. R is defined by: A R B if and only if $A \subseteq B$.
b. R is defined by: A R B if and only if if $A \cap B$ is not equal to NULL.
c. R is defined by: A R B if and only if $1 \in A \cap B$.

Solution:

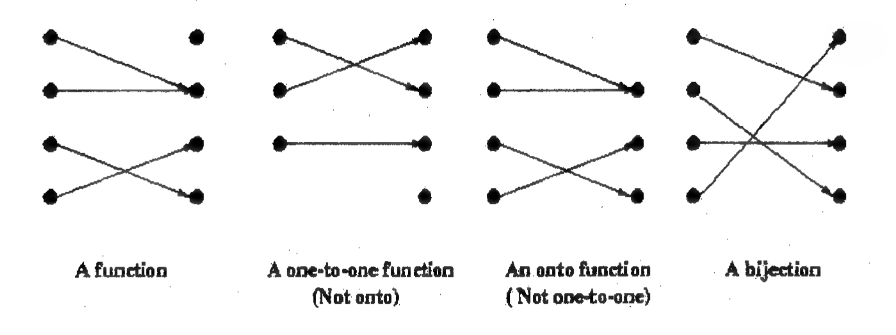
a. It is reflexive .It is NOT symmetric .It is transitive.

b. It is reflexive .It is symmetric. It is transitive.

c. It is NOT reflexive .It is symmetric .It is transitive.

4.1الاقتران

الاقتران هو علاقة تربط بين متغيرين او هدفين بحيث تعطي القيمة الواحدة(اواكثر من قيمة) من قيم المتغيرالاول قيمة واحدة فقط من قيم المتغير الثاني اي ان العلاقة بين المتغر الاول والثاني تكون اما من النوع واحد لواحد اوكثير لواحد وكما هو مبين في الشكل التالي:



A function | A one-to-one function (Not onto) | An onto function (Not one-to-one) | A bijection

وللمزيد من المعلومات عن الاقترانات يمكن الرجوع الى كتاب الرياضيات المنفصلة للمؤلفين.

5.1التعابير المنتظمة Regular expressions

التعبير المنتظم هو تمثيل للغة مؤلفة من مجموعة من الرموز بحيث تستخدم هذه المجموعة من قيل الالات المنتهية والتي سوف نستعرضها في هذه الكتاب سواء لقبولها او رفضها.

وفيما نورد اهم خصائص هذه التعابير:

- التعبير المؤلف من رمز واحد يشار الية بمجموعة مؤلفة من رمز واحد مثل:

`C` = {"c"}

- التعبير الفارغ هو التعبير الذي لا يحتوي على رموز:

$\varepsilon = \{``"\}$

- اتحاد تعبيرين هو تعبير يعبر عنه كما يلي:

$A+B = \{s \mid s \in A \text{ or } s \in B\}$

- دمج تعبيرين هو الاخر تعبير يمثل كما يلي:

$AB = \{ab \mid a \in A \text{ and } b \in B\}$

- تكرار تعبير هو الاخر تعبير يمثل كما يلي:

$A^* = \cup_{i\geq 0} A^i$ where $A^i = A\ldots A$ (i times)

$A^* = \{\varepsilon\} + A + AA + AAA + \ldots$

$A^+ = A + AA + AAA + \ldots = AA^*$

وفيما اهم الرموز المستخدمة لبناء التعابير المنتظمة:

Symbol	Stands for...
.	any single character
x*	x, zero or more times
x+	x, one or more times
x?	x once, or not at all(optional x)
x{n}	x exactly n times
x{n,m}	x, at least n but not more than m times
x|y	either x or y
xy	x followed by y

(x)	x as capturing group(more later)
[abc]	one of a or b or c, same as a|b|c
[^abc]	any character except a, b or c
[a-zA-Z]	a to z or A to Z(inclusive)

والجدول التالي يبين بعض الامثلة على التعابير المنتظمة:

RE	***Description***
$(0+1)^*111$	The set of strings containing only 0s and 1s that end in three consecutive 1s
$0^*1(0+1)^*$	The set of strings containing only 0s and 1s that have at least one 1
$0^*+0^*10^*$	The set of strings containing only 0s and 1s that have at most one 1
Σ^*	String of any characters
$\{a,\ldots,z,A,\ldots,Z\}(\{a,\ldots,z,A,\ldots,Z,0,\ldots,9,_\})^*$	The set of identifiers in Pascal
ΣΣΣ... Σ(80 times)	A line of 80 characters
1^+	A string of 1s, having at least one 1
$(\Sigma-\{a,e,i,o,u\})^*$	A string of letters not containing any vowel

وفيما يلي اهم القواعد المستخدمة للتعامل مع التعابير المنتظمة:

$\alpha+\alpha \equiv \alpha$

$\alpha+\phi \equiv \alpha$

$\alpha\cdot\phi \equiv \phi\cdot\alpha \equiv \phi$

$\varepsilon\cdot\alpha \equiv \alpha\cdot\varepsilon \equiv \alpha$

$(\varepsilon+\alpha)^* \equiv \alpha^*$

$\varepsilon+\alpha\alpha^* \equiv \alpha^*$

$\varepsilon+\alpha^*\alpha \equiv \alpha$

$(\alpha\beta)^*\alpha \equiv \alpha(\beta\alpha)^*$

$\alpha\alpha^* \equiv \alpha^*\alpha$

$(\alpha^*\beta)^*\alpha^* \equiv (\alpha+\beta)^*$

$\alpha^*(\beta\alpha^*)^* \equiv (\alpha+\beta)^*$

ويمكن استخدام هذه القواعد لاثبات تساوي التعابير المنتظمة وكما هو موضح في المثال التالي:

Prove that $0^* + 0^*1(\varepsilon+00^*1)^*\ 000^* = \varepsilon + (0 + 10)^*0$

$\text{LHS} = 0^* + 0^*1(\varepsilon+00^*1)^*\ 000^*$

$= (\varepsilon + 00^*) + 0^*1(00^*1)^*\ 000^*$

$= (\varepsilon + 00^*) + 0^*10(0^*10)^*\ 00^*$

$= \varepsilon + (\varepsilon + 0^*10(0^*10)^*)\ 00^*$

$= \varepsilon + (0^*10)^*\ 0^*0$

$= \varepsilon + (0+10)^*\ 0$

تستخدم لتوليد التعابير المتظمة او اللغة المؤلفة من محموعة من الرموز المنتهية مجموعة من القواعد grammar المحددة والمعرفة سابقا وتمثل هذه القواعد رياضيا كما يلي:

A grammar G =(V, T, P, S)

ويضم هذا النموذج:

- مجموعة منتهية من المتغيرات غيرالنهائية والتي يمكن ان تتفرع اتشكيل سلسلسل الرموز الخاصة باللغة.
- مجموعة منتهية من المتغيرات النهائية والتي لا تتفرع والتي تشكل الرموز الداخلة فب اللغة.
- مجموعة قواعد التوليد والتي تشكل تشكل عملية الاستدعاء الذاتي لتوليد الرموز.
- رمز البداية للغة اوللتعبير المنتظم.

مثال:

Example:

Terminal: a

Non-terminal: S

Productions: $S \rightarrow aS$

$S \rightarrow \varepsilon$

من المثال السابق وباستخدام هذه القاعدة يمكن توليد اي تكرار من الحرف المحدد.

مثال:

قاعدة توليد.{a|b}:

P: $S \rightarrow \varepsilon \mid a \mid b$

$S \rightarrow aSa$

$S \rightarrow bSb$

$G = (\{S\}, \{a,b\}, P, S)$

مثال:

قاعدة توليد لغة او تعبير منتظم مؤلف من سلسلة فيها عدد متساو من الاحرف a , b

$S \rightarrow aSbS \mid bSaS \mid \varepsilon$

وفيما يلي بعض الامثلة التوضيحية:

Example 1:

Terminal: a

Nonterminal: S

Productions: $S \rightarrow aS$

$S \rightarrow \varepsilon$

The derivation for a4 is:

S => aS

=> aaS

=> aaaS

=> aaaaS

Example 2:

Terminal: a

Nonterminal: S

Productions: $S \rightarrow SS$

$S \rightarrow a$

$S \rightarrow \varepsilon$

Derivation of a2 is as follows:

S => SS

=> SSS

=> SSa

=> SSSa

=> SaSa

=> εaSa

=> εaεa = aa

Example 3:

Terminals: a, b

Nonterminals: S

Productions:

$S \rightarrow aS$

$S \rightarrow bS$

$S \rightarrow a$

$S \rightarrow b$

More compact notation:

$S \rightarrow aS \mid bS \mid a \mid b$

Derive abbab as follows:

S => aS

=> abS

=> abbS

=> abbaS

=> abbab

CFL is(a+b)+

Example 4:

Terminals: a, b

Nonterminals: S, X

Productions:

$S \rightarrow XaaX$

$X \rightarrow aX \mid bX \mid \varepsilon$

CFL is(a + b)*aa(a + b)*

Derive abbaaba as follows:

S => XaaX

=> aXaaX

=> abXaaX

=> abbXaaX

=> abb_aaX = abbaaX

=> abbaabX

=> abbaabaX

=> abbaaba_ = abbaaba

هذا ويمكن تمثيل مجموعة الانتاج اوالتوليد بالهيكل الشجري وكما هو مبين في الامثلة التالية:

Example 1: CFG:

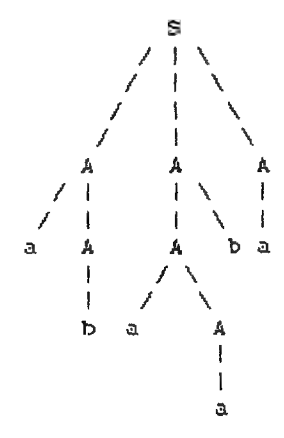
Terminals: a, b

Nonterminals: S, A

Productions:S → AAA | AA

A→ AA | aA | Ab | a | b

String abaaba has derivation tree:



Example2:

Terminals: a, b

Nonterminals: S

Productions: S → aS | SA | a

The word aa can be generated by two different trees:

S S

/ \ / \

a S S a

| |

a a

Example 3:

Terminals: a, b

Nonterminals: S, X

Productions: S → aS | aSb | X

X → Xa | a

The word aa has two different derivations that correspond to different syntax trees:

1. S => aS => aX → aa

S

/ \

a S

|

X

|

a

الوحدة الثانية

آلة الحالة المنتهية

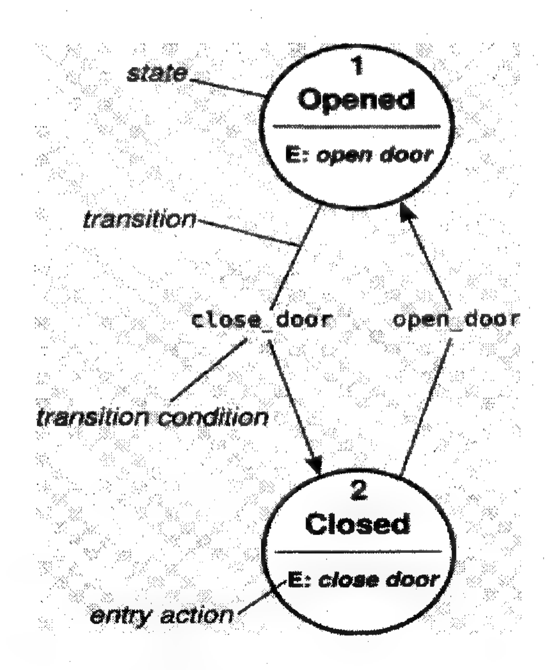
Finite state machine(FSM)

2

1.2 مقدمة

الة الحالة المنتهية Afinite state machine(FSM) or finite state automaton or state machine هي نموذج يبين سلوك الوحدات الذاتية المخصصة لحل مشكلة معينة وذلك اعتمادا على مجموعة من العوامل الداخلة في النموذج مثل مجموعة الحالات التي تقع فيها الالة ومجموعة النتقالات من حالة الى اخرى ومجموعة الافعال اوالاحداث المولدة نتيجة لعملية الانتقال من حالة الى اخرى وبتاثير المدخلات المستخدمة في الالة.

ويبين الشكل التالي نموذجا اومخططا لالة الحالة والتي يمكن التعبير عنها بمجموعة العوامل التالية:



- مجموعة المدخلات.
- مجموعة الحالات ومن بيتها الحالة الابتدائية.
- مجموعة المخرجات.

- دالة الانتقال والتي تفيد بنقل الالة من حالة محددة الى حالة اخرى محددة اعتمادا على البيانات المتوفرة حاليا.

تسخدم الحالة لتخزين معلومات عن سلوك الالة في الماضي وهي تعكس التغيرات الناجمة في الالة نتيجة لقراءة اومعالجة مجموعة من المدخلات.

ترتبط عملية انتقال الالة من حالة الى اخرى بشرط اواكثر و عادة ما يرتبط الشرط بقيمة الحالة الحالية وقيم المدخلات الحالية ويتم التعبير عن دالة النتقال من خلال جدول يشبه الى حد ما الجدول المبين ادناه:

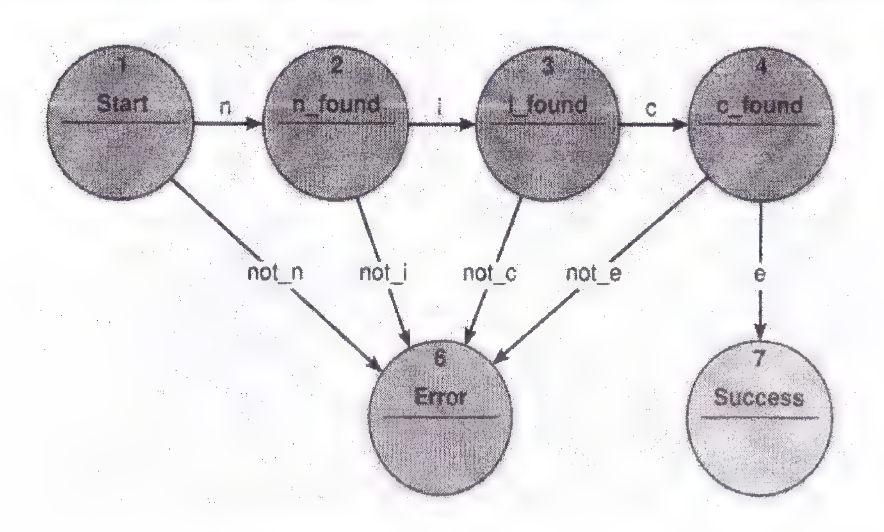
State transition table

Current State -> Condition	State A	State B	State C
Condition X	...	...	...
Condition Y	...	State C	...
Condition Z	...	...	...

تستخدم الالات المنتهية في كثير من التطبيقات وبشكل عام تصنف الالات المنتهية الى:

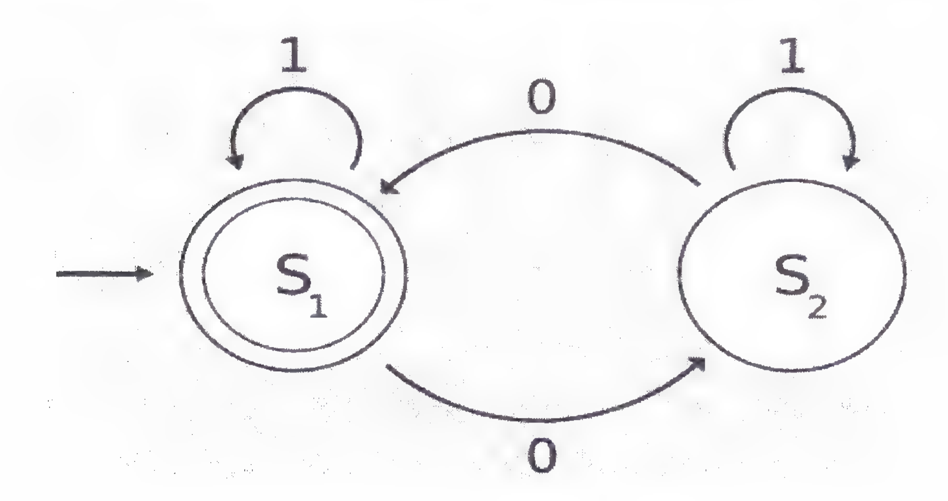
1. الالة المميزة Recognizer

وتستخدم هذه الالة في الغالب لتمييز مجموعة من المدخلات والتعرف عليها او بمعنى اخر التعرف على نمط معين من البيانات تشكل جزءا من المدخلات والمثال التالي يبين مخطط الحالات لالة منتهية تعمل على تمييز الكلمة nice

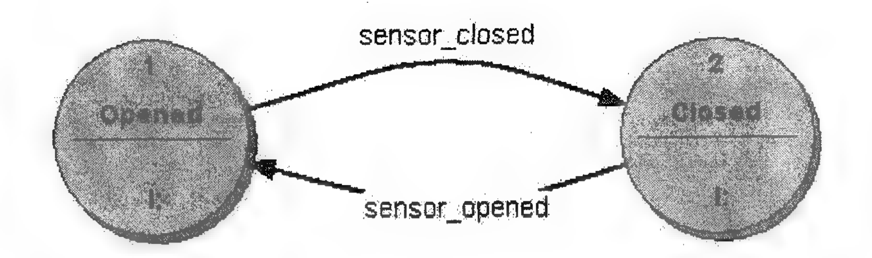


2. الالة المنتهية التي تقبل مجموعة من المدخلات Acceptors

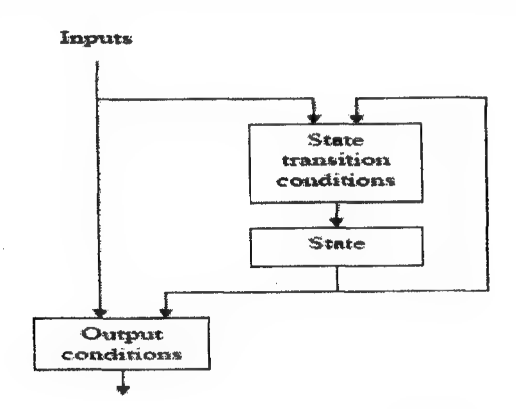
ويقبل هذا النوع من الالات مجموعة من المدخلات والمثال التالي يبين مخطط الة منتهية تحدد فيما اذا كان الرقم الثنائي زوجي ام فردي.



ومن الامثلة على الالات المنتهية المتحسسات والتي يمكن ان تتاثر بالمدخلات لتبقى في نفس الحالة او تنتقل الى حالة جديدة وكما هو مبين في الشكل التالي:



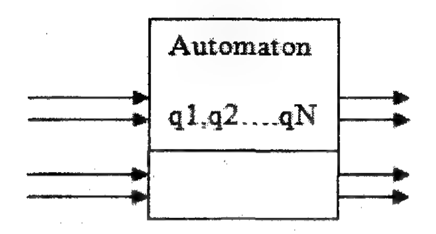
وبشكل عام فان الالة المتهية ما هي الا وحدة تقبل مجموعة من المدخلات وتنتقل من حالة الى حالة اخرى او يمكن ان تبقى في نفس الحالة اعتمادا على قيم المدخلات والحالة الحالية وعند الانتقال فانها تعمل على توليد بعض المخرجات ويبين الشكل التالي اهم مكونات الالة المتهية اعتمادا على هذا التعريف:



2.2 النموذج الرياضي للالة المنتهية:

كما اشرنا سابقا فان الالة المنتهية ما هي الا وحدة تحكم تحتفظ بمعلومات عن سلوك الالة في الماضي بناء على المدخلات التي تمت قراءتها وتشكل مجموعة الحالات حالة الالة المنتهية واعتمادا على الحالة الحالية والبيانات الحالية المقروءة فان الالة يمكن ان تنتقل الى حالة جديدة او تبقى في نفس الحالة

منتجة بذلك الخرجات اذا لزم الامر او تطلب الامر من ان تقوم الوحدة الذاتية بانتاج مخرجات والشكل التالي يبين نموذج الالة المنتهية:



وبناء على ما تقدم يمكن وصف الالة المنتهية بما يلي:

- مجموعة المدخلات.
- مجموعة المخرجات(ان وجدت).
- مجموعة الحالات بما فيها الحالة الابتدائية والحلات النهائية.
- دالة الانتقال.

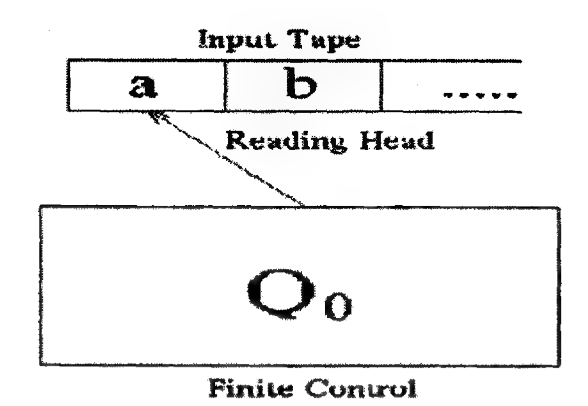
هناك نوعان من الالات المنتهية:

1. الالة المنهية المحدودة.
2. الالة المنتهية غير المحدودة.

وفيما سوف نستعرض هذين النوعين.

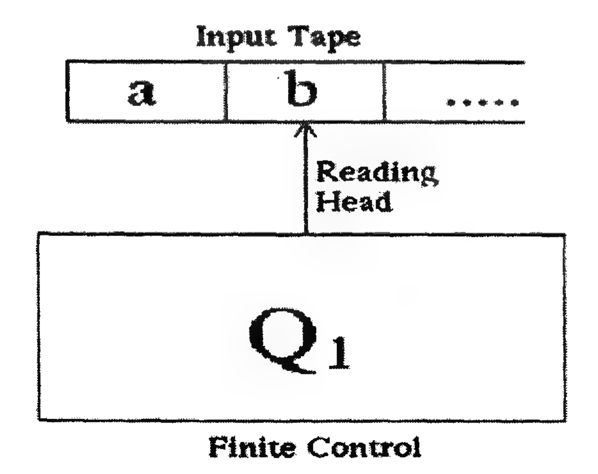
1. الالة المنهية المحدودة Deterministic Finite Automata

يمكن تصور هذه الالة كما هو مبين في الشكل التالي على انها وحدة مؤلفة من ما يلي:



- شريط المدخلات والمؤلف من مجموعة من الرموز.
- وحدة التحكم والتي تحتفظ بمجموعة الحالات.
- راس القراءة والكتابة والمخصص فقط للقراءة(قراءة المدخلات بدون كتابة المخرجات).
- يتحرك راس القراءة فقط باتجاه اليمين وبعد كل عملية قراءة ينتقل راس القراءة لموقع.

واحد فقط باتجاه اليمين وكما هو مبين في الشكل التالي:



تمتلك الة الحالة المحدودة حالة نهائية اواكثر ويجب ان ينتهي تنفيذها في احدى الحالات النهائية.

يتم وصف الالة المنتهية رياضيا بالنموذج التالي:

$$FA= <Q,I,\delta,q0,F>$$

والذي يشمل:

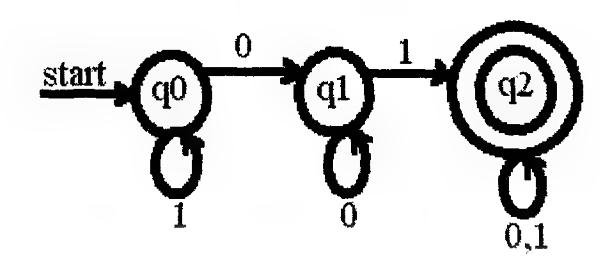
- مجموعة محدودة من الحالات.
- مجموعة منتهية من رموز المدخلات.
- دالة الانتقال والتي تاخذ المعاملات الممثلة بالحالة الحالية والرمز المقروء للانتقال الى الحالة الجديدة.
- الحالة الابتدائية.
- مجموعة الحالات المقبولة اومجموعة الحالات النهائية والمنتمية الى مجموعة الحالات الكلية.

يتم في المخطط التعبير عن الحالة بالائرة اما الأنتقال من حالة الى اخرى فيعبر عنها السهم على ان يكتب عليه الرمز المقروء.

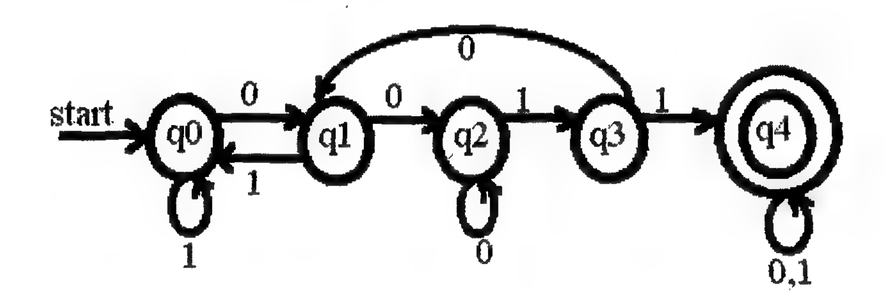
تستخدم الالات الننتهية المحدودة لتمييز النماط في سلسلة الرموز او تعمل على اكتشاف تسلسل معين لمجموعة من الرموز في سيل المدخلات.

امثلة:

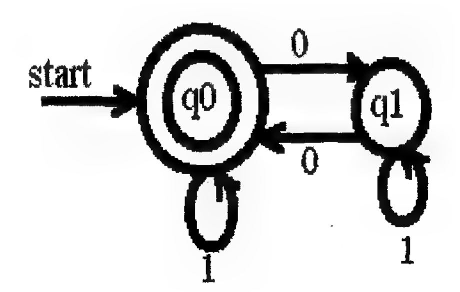
1. ابن الالة المنتهية المحدودة والتي تعمل على اكتشاف الصفر متبوعا بالواحد في مجموعة من المدخلات المؤلفة من الصفر والواحد:



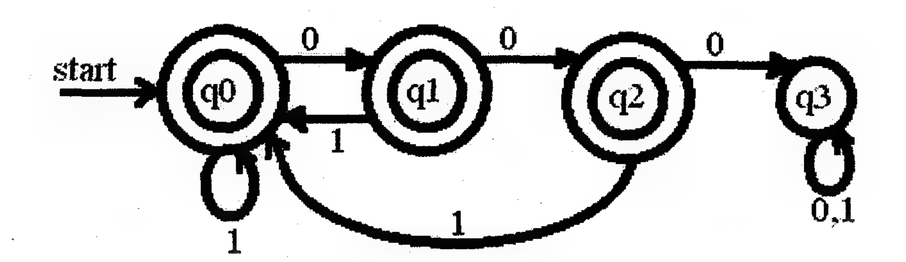
2. ابن الالة المنتهية المحدودة والتي تعمل على اكتشاف صفرين متتابعين متبوعين بواحدين متتابعين(اي اكتشاف السلسلة الجزئية 0011).



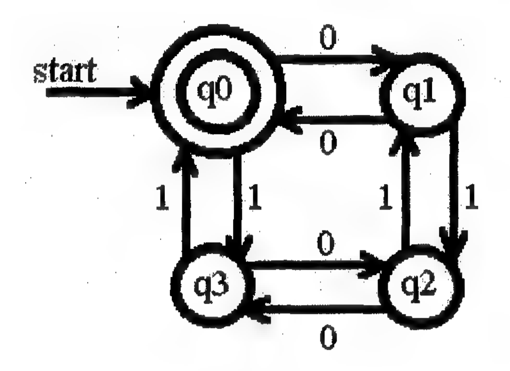
3. ابن الالة المنتهية والتي تقبل سلسلة من الرموز مؤلفة من عدد زوجي من الاصفار واي عدد من الوحدات:



4. ابن الالة المنتهية والتي تقبل كافة مجموعة الرموز باستثناء 3 وحدات متتابعة:



5. ابـن الالـة المنتهيـة الـتي تقبـل عـدد زوجـي مـن الاصـفار وعـدد زوجـي مـن الوحدات.



6. ابن الة الحالة المنتهية والتي تحقق العلاقة: باقي قسمة س على 5 = 2

لاحظ هنا ان مجموعة المدخلات التي تقود الى الحالة النهائية هي:

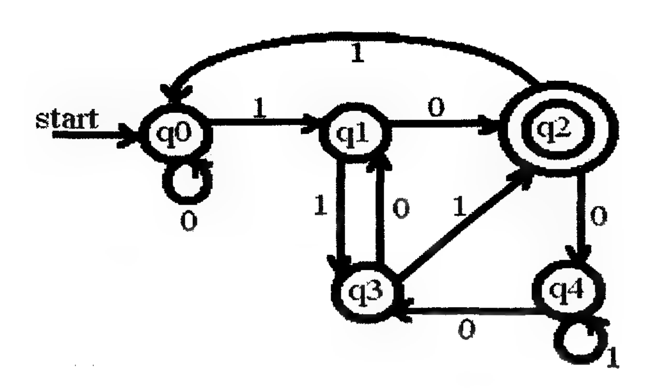
10

111

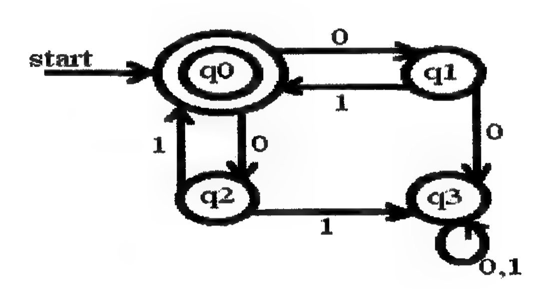
1100

10110

وغيرها الكثير وحسب ما هو مبين في مخطط الالة التالي:



7. ابن مخطط الة الحالة المنتهية التي تقبل تكرار الصفر والواحد $(0+1)^*$ بعدد متساو من الاصفار والوحدات علما بان كل بداية يجب ان تحتوي على الاكثر على صفر اضافي عن الوحدات او على الاكثر واحد اضافي عن عدد الاصفار:



مما تقدم يمكن النظر الى الة الحالة المنتهية كمعدات(وسوف نتطرق الى هذا لاحقا في هذا الكتاب ان شاء الله) مكونة من الاجزاء التالية:-

- مسجل داخلي.
- مجموعة من القيم التي تكتب في المسجل.
- شريط الرموز.
- راس القراءة.
- مجموعة من التعليمات والممثلة لدالة الانتقال.

3.2 اهم المصطلحات:

1. اللغة المقبولة من قبل الالة المنتهية هي مجموعة الرموز المقروءة والتي تؤدي قراءتها الى الأنتقال من الحالة الابتدائية الى احدى الحالات النهائية.

فمثلا اذا كانت مجموعة الرموز مؤلفة من:

$\Sigma = \{a, b\}$

فهناك مجموعة من الحالات لبناء الالات المنتهية والتي تقب اللغات المؤلفة من نماذج السلسل الحرفية التالية:

Strings

a	
ab	$u = ab$
$abba$	$v = bbbaaa$
$baba$	$w = abba$
$aaabbbaabab$	

2. تطبق على السلاسل الرمزية مجموعة من العمليات اهمها الدمج والقراءة العكسية وتحديد طول السلسلة الرمزية كما هو مبين في الامثلة التالية:

$w = a_1 a_2 \cdots a_n$ $v = b_1 b_2 \cdots b_m$

$w = abba$ $v = bbbaaa$

Concatenation $wv = a_1 a_2 \cdots a_n b_1 b_2 \cdots b_m$

$wv = abbabbbaaa$

Reverse $v^R = b_m \cdots b_2 b_1$

$v^R = aaabbb$

$w = a_1 a_2 \cdots a_n$

Length: $|w| = n$

Examples: $|abba| = 4$

$|aa| = 2$

$|a| = 1$

For any letter: $|a| = 1$

For any string wa : $|wa| = |w| + 1$

Example:
$$\begin{aligned} |abba| &= |abb| + 1 \\ &= |ab| + 1 + 1 \\ &= |a| + 1 + 1 + 1 \\ &= 1 + 1 + 1 + 1 \\ &= 4 \end{aligned}$$

$$|uv| = |u| + |v|$$

Example: $u = aab, \quad |u| = 3$

$$v = abaab, \quad |v| = 5$$

$$|uv| = |aababaab| = 8$$

$$|uv| = |u| + |v| = 3 + 5 = 8$$

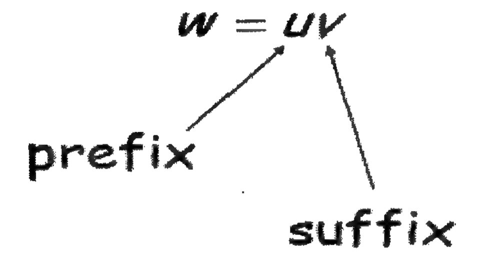
3. السلسلة بدون رموز هي سلسلة فارغة ويمكن استخدامها في بعض الاحيان لنقل الالة من حالة الى اخرى دون الحاجة الى مدخلات:

$$|\lambda| = 0$$

$$\lambda w = w\lambda = w$$

$$\lambda abba = abba\lambda = abba$$

4. اذا كانت $w=uv$ مجموعة الرموز المقبولة من قبل الالة المنتهية فان السلسلة الاولى تنفذ اولا وتدعى سلسلة البداية اما السلسلة الثانية فتنفذ ثانيا وتدعى السلسلة البعدية



واذا ضمت السلسلة مجموعة الاحرف abbab فانها ستنفذ بقراءة حرف حرف ومن اليسار الى اليمين كما يلي:

Prefixes	Suffixes
λ	$abbab$
a	$bbab$
ab	bab
abb	ab
$abba$	b
$abbab$	λ

5. قد تتكرر عملية قراءة مجموعة الرموز وفي هذه الاحالة تستخدم السلسة الرمزية مرفوعة لقوة تساوي عدد مرات التكرار وكما هو مبين ادناه:

$$w^n = \underbrace{ww \cdots w}_{n}$$

$$(abba)^2 = abbaabba$$

for any :

$$w \qquad w^0 = \lambda$$

$$(abba)^0 = \lambda$$

6. تستخدم النجمة مع مجموعة الرموز للاشارة الى الى كافة السلاسل الرمزية التي يمكن تكوينها من السلسلة الرمزية بما في ذلك السلسلة الرمزية الفالرغة اما اشارة الزائد فتشير الى كافة السلاسل الرمزية التي يمكن الحصول عليها من السلسلة الاصلية باستثناء السلسلة الفارغة:

Example:

$$\Sigma = \{a, b\}$$

$$\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, aab, \ldots\}$$

$$\Sigma^+ = \Sigma^* - \lambda$$

$$\Sigma^+ = \{a, b, aa, ab, ba, bb, aaa, aab, \ldots\}$$

7. اللغة المقبولة من الة الحالة المنتهية عي سلسلسة جزئية تنتمي الى مجموعة السلاسل الرمزية التي يمكن تكوينها من مجموعة الرموز:

Examples:

$$\Sigma = \{a, b\}$$

$$\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, aab, \ldots\}$$

Language L(M)=

$$\{\lambda\}, \quad \{a, aa, aab\}$$

$$\{\lambda, abba, baba, aa, ab, aaaaaa\}$$

8. اللغة غير المنتهية هي مجموعة الرموز المقروءة وغير المنتهية خاصة عندما يكون هناك تكرار في الة الحالة المنتهية من خلال الوصول الى حالة نهائية ثم الخروج منها والعودة اليها:

An infinite language

$$L = \{a^n b^n : n \geq 0\}$$

$$\left.\begin{array}{l} \lambda \\ ab \\ aabb \\ aaaaabbbbb \end{array}\right] \in L \qquad abb \notin L$$

9. تنفذ على اللغات المقبولة من الالات النتهية مجموعة من العمليات اهمها الاتحاد والتقاطع والفرق والمكمل تماما كما تنفذ هذه العمليات على المجموعات:

The usual set operations

$$\{a, ab, aaaa\} \cup \{bb, ab\} = \{a, ab, bb, aaaa\}$$
$$\{a, ab, aaaa\} \cap \{bb, ab\} = \{ab\}$$
$$\{a, ab, aaaa\} - \{bb, ab\} = \{a, aaaa\}$$

Complement: $\overline{L} = \Sigma^* - L$

$$\overline{\{a, ba\}} = \{\lambda, b, aa, ab, bb, aaa, \ldots\}$$

10. اللغة المعكوسة تؤخذ من مجموعة الرموز المشكلة للغة الاصلية وتقرا بشكل معكوس:

Definition: $L^R = \{w^R : w \in L\}$

Examples:

$$\{ab, aab, baba\}^R = \{ba, baa, abab\}$$

$$L = \{a^n b^n : n \geq 0\}$$

$$L^R = \{b^n a^n : n \geq 0\}$$

11. ناتج دمج لغتين هو لغة مقدمتها من اللغة الاولى ونهايتها من اللغة الثانية:

Definition:

$$L_1L_2 = \{xy : x \in L_1, y \in L_2\}$$

Example:

$$\{a, ab, ba\}\{b, aa\}$$

$$= \{ab, aaa, abb, abaa, bab, baaa\}$$

هذا ويمكن دمج اللغة الواحدة لتكرار قراءتها اكثر من مرة:

Definition:

$$L^n = \underbrace{LL\cdots L}_{n}$$

Example:

$$\{a,b\}^3 = \{a,b\}\{a,b\}\{a,b\} =$$
$$\{aaa, aab, aba, abb, baa, bab, bba, bbb\}$$

Special case:

$$L^0 = \{\lambda\}$$

$$\{a, bba, aaa\}^0 = \{\lambda\}$$

Example:

$$L = \{a^n b^n : n \geq 0\}$$

$$L^2 = \{a^n b^n a^m b^m : n, m \geq 0\}$$

$$aabbaaabbb \in L^2$$

12. التغطية او تغطية النجمة هي مجموعة اللغات التي يمكن تغطيتها اوالوصول اليها من لغة محددة:

Star-Closure (Kleene *)

Definition:

$$L* = L^0 \cup L^1 \cup L^2 \cdots$$

Example:

$$\{a, bb\}^* = \left\{ \begin{array}{l} \lambda, \\ a, bb, \\ aa, abb, bba, bbbb, \\ aaa, aabb, abba, abbbb, \ldots \end{array} \right\}$$

اما التغطية الموجبة في التغطية الكلية للغة مستثنيا منها مجموعة الرموز الفارغة:

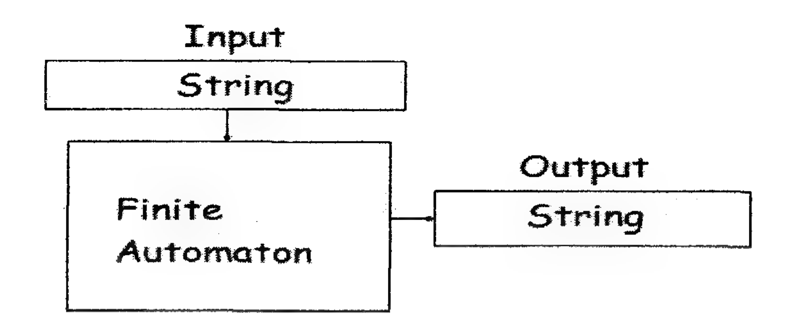
Positive Closure

Definition: $$L^{+} = L^{1} \cup L^{2} \cup \cdots$$

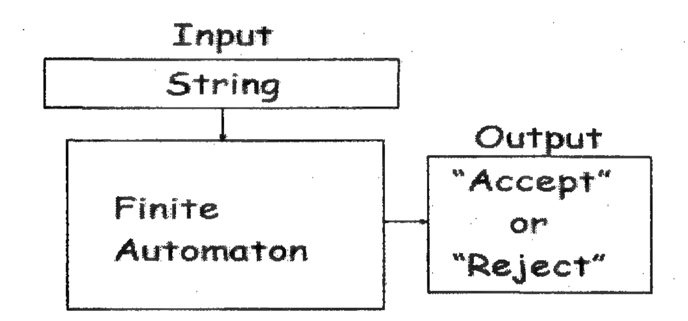
$$= L^{*} - \{\lambda\}$$

$$\{a, bb\}^{+} = \left\{ \begin{array}{l} a, bb, \\ aa, abb, bba, bbbb, \\ aaa, aabb, abba, abbbb, \ldots \end{array} \right\}$$

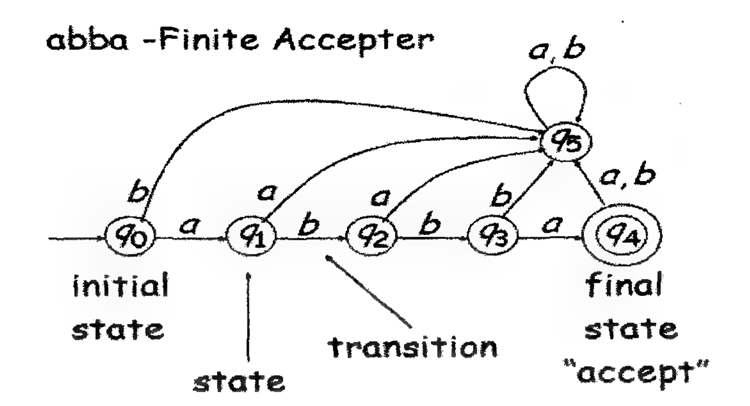
13. تصنف الالة المنتهية الى الالة المنتجة للمخرجات حيث يتم في هذا النوع من الالات الحالة المنتهية انتاج المخرجات وتعتمد قيمة الخرج المنتج على الحالة الحالية للالة وقيمة الدخل المقروء وكذلك الحال بالنسبة للحالة القادمة والتي تعامل كاقتران يعتمد على الدخل والحالة الحالية كما هو الحال في هذا وسوف نستعرض هذا النوع من الالات لاحقا وسوف نستعرض نماذج من هذه الالات مثل الة موور والتي يعتمد فيها الخرج على الدخل والحالة الحالية والة ميلي والتي يعتمد فيها الخرج على الحالة الحالية ومن اشهر انواع المعدات الممثلة لهذا النوع من الالات المتحسسات واشكل التالي يبين المخطط الصندوقي للالة المنتهية المنتجة للمخرجات:



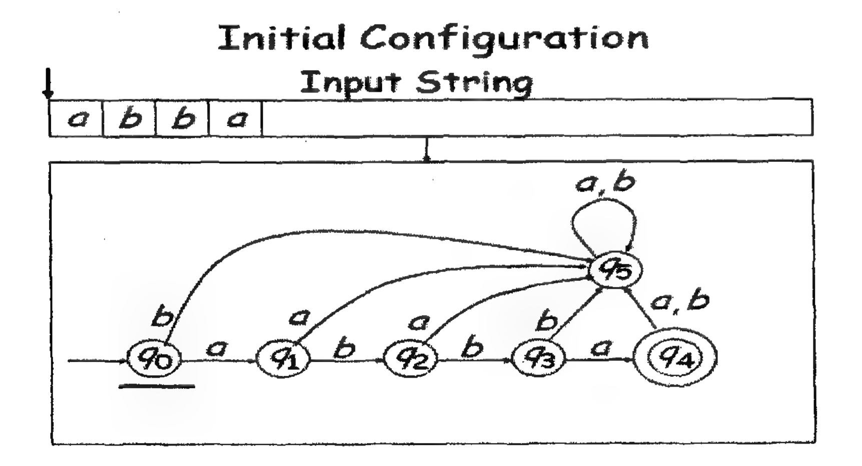
اما النوع الثاني من الالات المنتهية فهو الذي نحن بصدده في هذه الوحدة الا وهو الالة المميزة وهي الة حالة منتهية تعمل على تمييز مجموعة من الرموز تشكل لغة الالة بحيث تقود قراءتها الى حالة نهائية اوتعمل على رفض مجموعة من الرموز والتي تقود قراءتها الى الانتقال الى حالة غير نهائية والشكل التالي يبين هذا النوع من الالات:



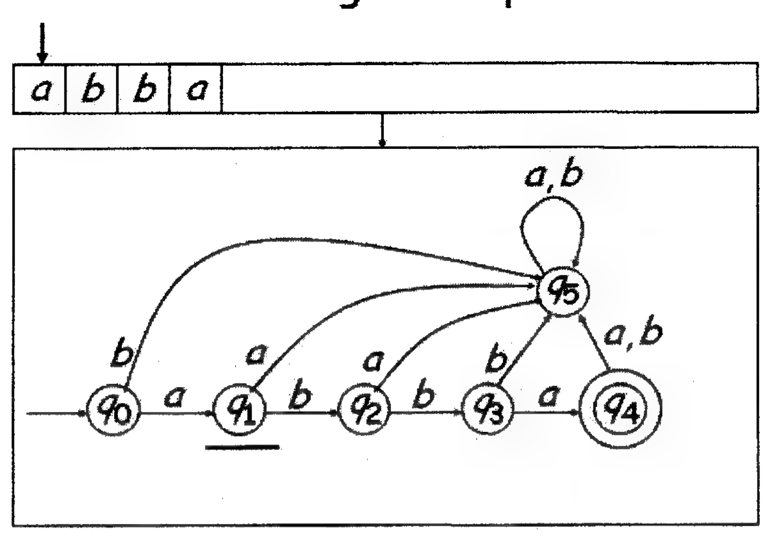
14. تستخدم مجموعة من الرموز لتمثيل الالة المنتهية بواسطة مخطط الحالات وهذه الرموز مبينة في الشكل التالي:

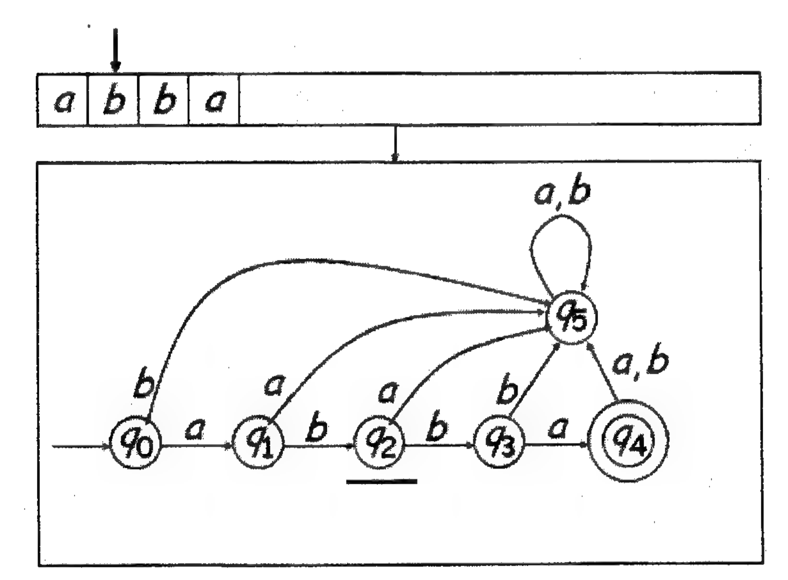


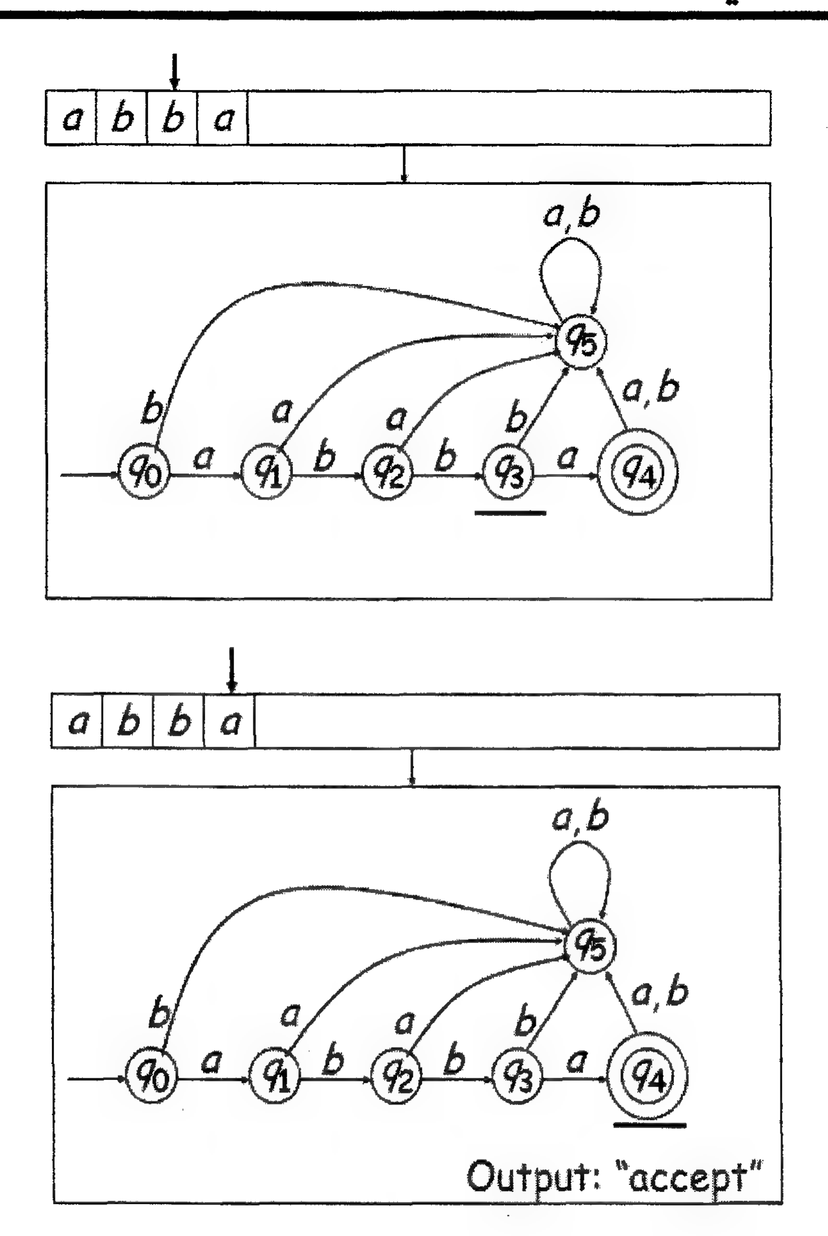
15. تعبر هيئة الالة Configuration عن الحالة الحالية ومجموعة الرموز الـت لم تقـرا بعـد ويمكـن للالـة ان تنتقـل مـن هيئـة الى اخـرى بعـد قـراءة رمـز مـن مجموعـة الرمـوز المـسجلة علـى الـشريط هـذا ويمكـن التعـبير عـن الهيئـة باسـتخدام المخطـط اوباسـتخدام عمليـة التمثيـل الرياضـي وفيمـا يلـي مجموعة من الاشكال والتي تبين كيفية النتقال من هيئة لاخرى في حالة قبول اللغة اوتمييز مجموعة من الرموز:



Reading the Input

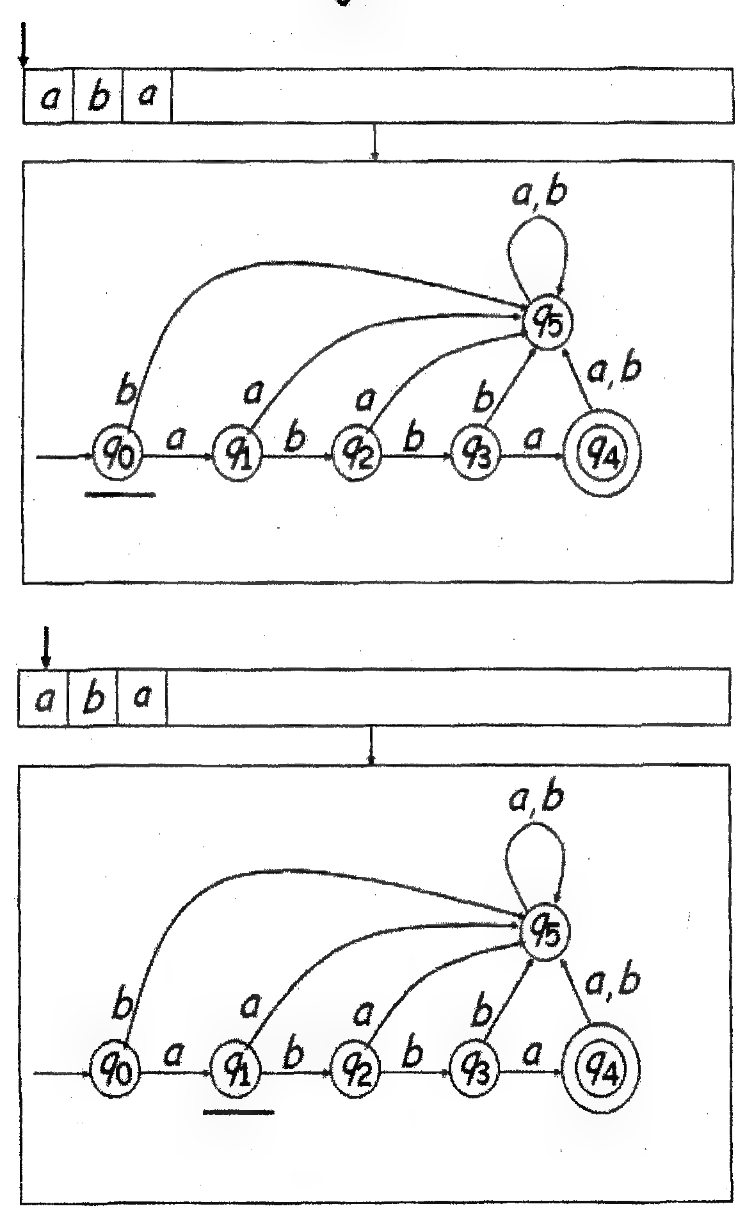


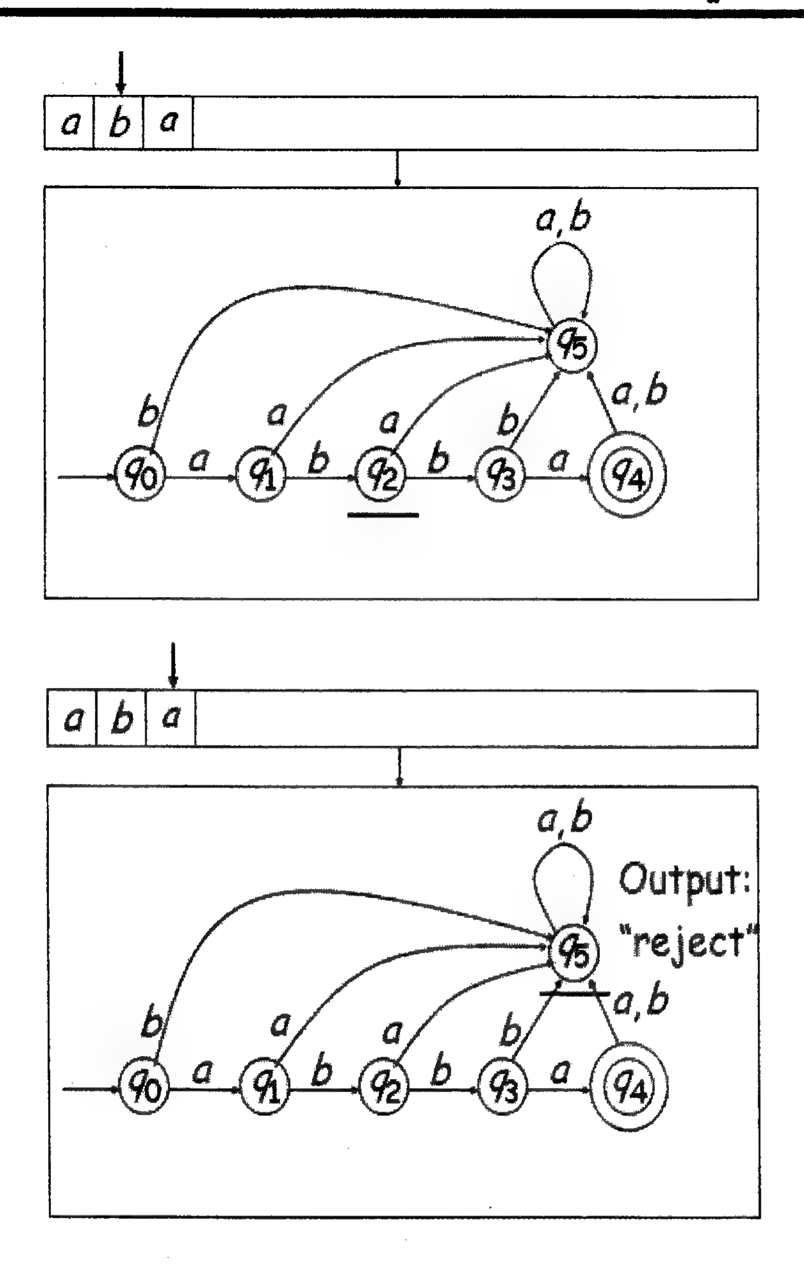



a b b a
a,b
b
a
a
b
a,b
a
b
b
a
a b b a
a,b
b
a
a
b
a,b
a
b
b
a
Output: "accept"

والخطوات التالية تبين عملية رفض اللغة او عدم قبولها:

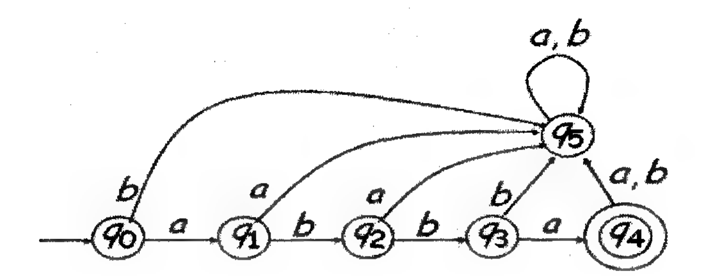
Rejection



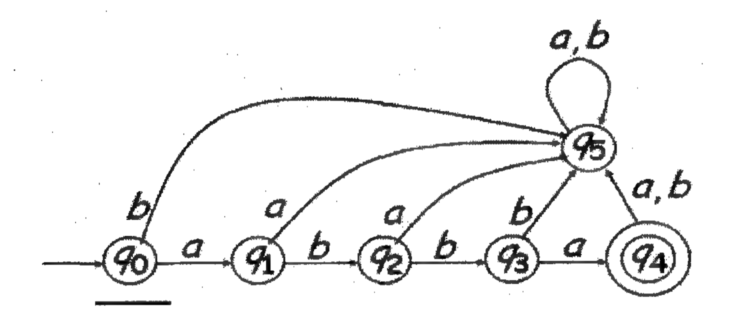


16. يتم وصف الة الحالة المنتهية باستخدام مخطط الحالات او جدول الانتقال او الهيئة وفيما يلي بعض الامثلة التوضيحية على تمثيل الالة المنتهية:

$\Sigma = \{a, b\}$

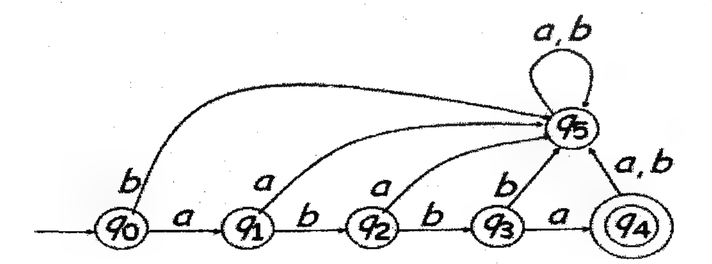


Initial State q_0



Set of Final States F

$F = \{q_4\}$

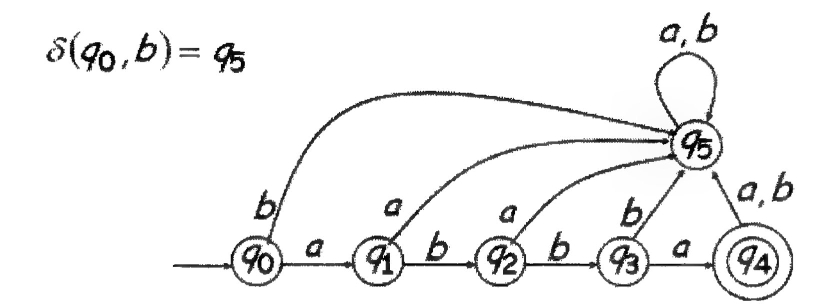


Transition Function δ

$$\delta : Q \times \Sigma \rightarrow Q$$

$$\delta(q_0, a) = q_1$$

$$\delta(q_0, b) = q_5$$



Transition Function δ

δ	a	b
q_0	q_1	q_5
q_1	q_5	q_2
q_2	q_2	q_3
q_3	q_4	q_5
q_4	q_5	q_5
q_5	q_5	q_5

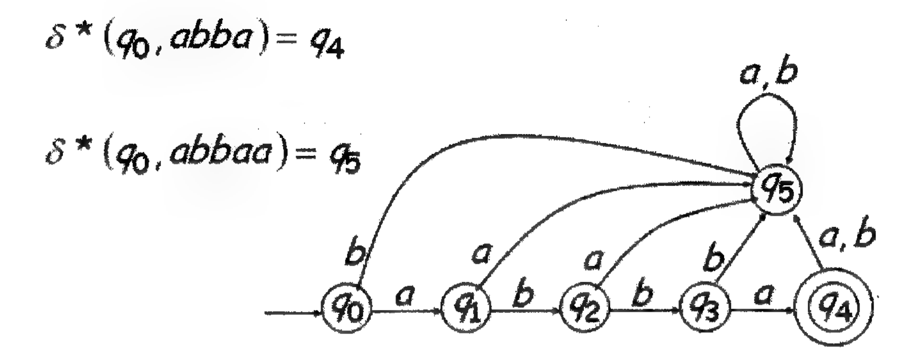
Extended Transition Function δ^*

$$\delta^* : Q \times \Sigma^* \to Q$$

$\delta^*(q_0, ab) = q_2$

$\delta^*(q_0, abba) = q_4$

$\delta^*(q_0, abbaa) = q_5$



Recursive Definition

$$\delta^*(q, \lambda) = q$$

$$\delta^*(q, wa) = \delta(\delta^*(q, w), a)$$

$\delta^*(q_0, ab) =$

$\delta(\delta^*(q_0, a), b) =$

$\delta(\delta(\delta^*(q_0, \lambda), a), b) =$

$\delta(\delta(q_0, a), b) =$

$\delta(q_1, b) =$

q_2

4.2 اللغة المقبولة من الة الحالة المنتهية

تكتب دالة الانتقال بالصورة التالية:

$$\delta(q, a) = p$$

والتي تعني النتقال من الحالة q الى الحالة p عند قراءة الرمز او الحرف a
وهناك اصطلاح اخر لوصف هيئة الالة ويكتب هذا الاصطلاح كما يلي:

$$[q_i, aw] \vdash [q_j, w]$$

حيث يشير الطرف الايسر الى الحالة الحالية والرمز الذي يقف عنده راس القراءة ومجموعة الرموز المتبقية والتي لم تقرا بعد اما الطرف الايمن فيشير الى الحالة بعد قراءة الرمز ومجموعة الرموز المتبقية.

والصورة السابقة للهيئة هي مكافئة للصيغة التالية:

$$[q_i, aw] \vdash [\delta(q_i, a), w], \text{where } \delta(q_i, a) = q_j$$

تقبل الة الحالة المنتهية مجموعة الرموز:

$$w = a1a2\ldots an$$

اذا كان هناك تتابع من الانتقالات بحيث:

- تبدء بحالة ابتدائية.
- تنتهي بحالة مقبولة.
- تمتلك مجموعة من الرموز من المجموعة الكلية للرموز مساوي ل

$$a1, a2, \ldots, an = w.$$

هذا ويمكن توسعة دالة الانتقال لتشير الى الحالة النهائية كما يلي:

"δ-hat"(q,w),

والتي تعني نقل الالة من الحالة المحددة بقراءة مجموعة الرموز المحددة واذا كانت مجموعة الرموز خالية فان الانتقال يتم فوريا وهنا يمكن ان نستخدم لامدا او ايبسلون للتعبير عن عملية النتقال دون الحاجة الى وجود مدخلات:

If $|w| = 0$, then "δ-hat"$(q,\lambda) = q$

اما اذا كانت مجموعة الرموز تشكل رمزا واحدا فتكتب دالة الانتقال كما يلي:

If $|w| = 1$, then "δ-hat"$(q, a) = \delta(q, a)$.

اما اذا كانت مجموعة الرموز اكث من رمز فيمكن التعبير عن دالة الانتقال كما يلي:

Let $|w|$ be $n > 1$.

Then $w = ua$ and "δ-hat"$(q, ua) = \delta($"δ-hat"$(q, u), a)$, where a is a single symbol

وبهذا فانه للالة المنتهية التالية:

$M = (Q,\Sigma,\delta, q_0, F)$

تكون مجموعة الرموز المقبولة:

String w if "δ–hat" (q_0, w) is in F.

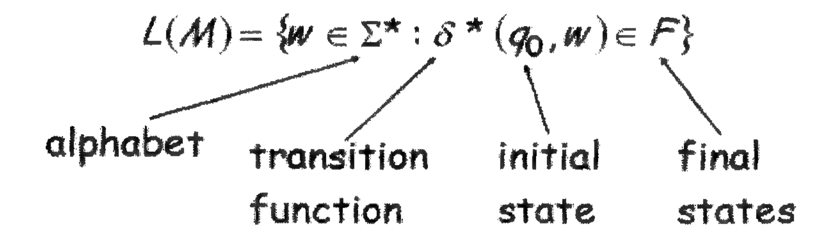
اما اللغة المقبولة من الالة فهي مجموعة الرموز المقروءة والتي تؤدي فراءتها لنقل الالة من حالة الى حالة مقبولة او نهائية ويعبر عن اللغة كما يلي:

L (M) = {w | "δ–hat" (q_0, w) is in F}.

وفيما يلي التعريف الرياضي للغة المقبولة من قبل الوحدة المنتهية:

For a DFA $M = (Q, \Sigma, \delta, q_0, F)$

Language accepted by M :



يمكن استخدام ايضا جدول الانتقال او دالة الانتقال لتمثيل الة الحالة المنتهية ولناخذ المثال التوضيحي التالي:

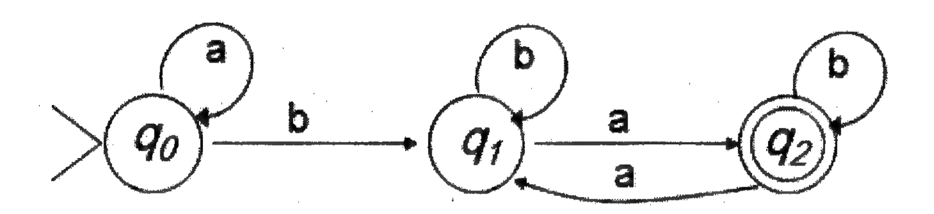
استخدم البيانات التالية لرسم مخطط الحالات للالة المنتهية:

- Let M = {Q, Σ, δ, q_0, F} be a DFA.

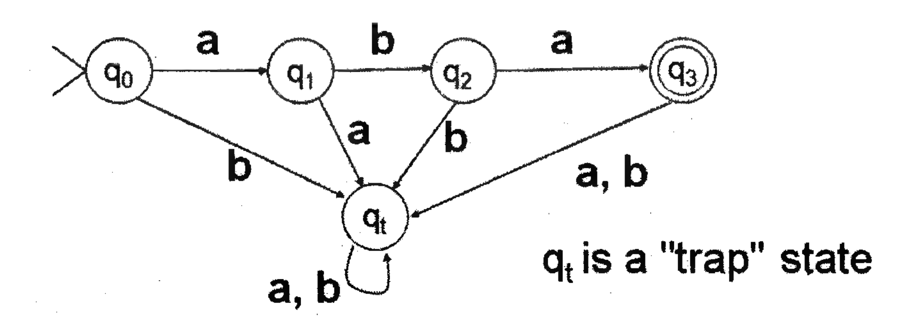
  $Q = \{q_0, q_1, q_2\}$
  $\Sigma = \{a, b\}$
  $F = \{q_2\}$

δ	a	b
$>q_0$	q_0	q_1
q_1	q_2	q_1
$*q_2$	q_1	q_2

الحل:



كما اشرنا سابقا فان اللغة المقبولة من الالة هي مجموعة الرموز التي تؤدي قراءتها الى الانتقال الى حالة نهائية وفي بعض الاحيان فان بعض الرموز يمكن ان تستثنى من اللغة وفي هذه الحالة تؤدي قراءة مثل هذه الرموز الى الوقوع في حالة تسمى حالة الفخ اوحالة ما يسمى رفض الرموز وكما هو مبين في الشكل التالي:



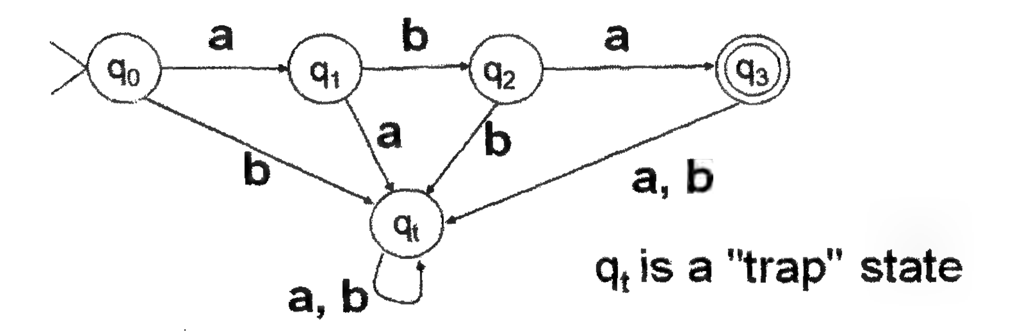
ولإيجاد اللغة المقبولة من قيل الة الحالة المنتهية يمكن اتباع الخطوات التالية:

- اوجد التعابير المنتظمة لمجموعة الرموز التي تؤدي قراءتها u1,...,un للانتقال من الحالة الابتدائية الة حالة نهائية.
- اوجد التعابير المنتظمة لكافة مسارات الخروج من كل حالة نهائية والرجوع اليها v1,...,vm
- ستكون عند هذا الرموز المقبولة هي:

$(u_1 \cup \ldots \cup u_n)(v_1 \cup \ldots \cup v_m)^*$

مثال:

اوجد اللغة المقبولة من قبل الة الحالة المنتهية الممثلة بالمخطط التالي:



الحل:

مجموعة الرموز التي تقود الى حالة نهائية من الحالة الابتدائية هي aba

وعليه فان اللغة المقبولة من قبل هذه الالة هي:

L (M) = aba

مثال:

ابن مخطط الحالات للالة المعبر عنها بما يلي ثم اوجد اللغة المقبولة من هذه الالة:

Let $M = \{Q, \Sigma, \delta, q_0, F\}$ be a DFA.

$Q = \{q_0, q_1, q_2\}$

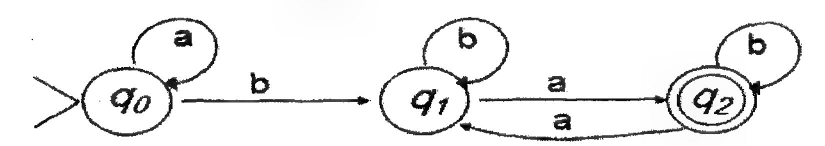
$\Sigma = \{a, b\}$

$F = \{q_2\}$

δ	a	b
$>q_0$	q_0	q_1
q_1	q_2	q_1
$*q_2$	q_1	q_2

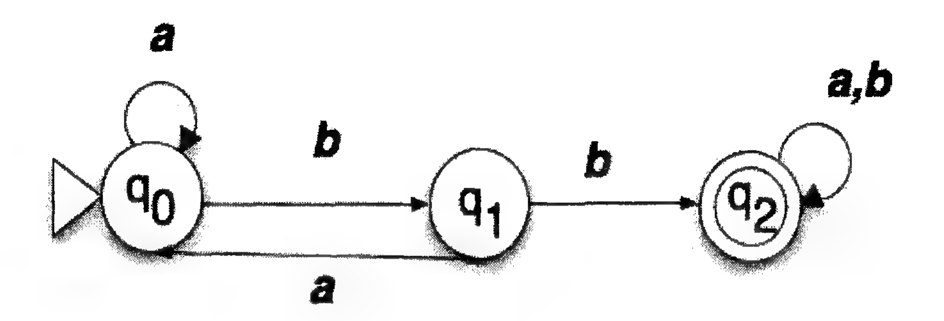
الحل:

$RE = a^*b^+a(b \cup ab^*a)^*$



مثال:

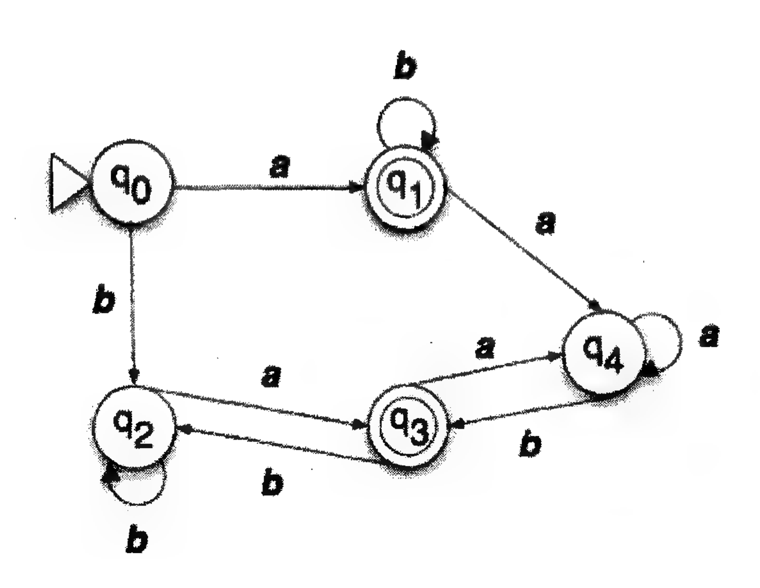
اوجد اللغة المقبولة للالة التالية:



$$a^*b(aa^*b)^*b(a \cup b)^* = a^*b(a^+b)^*b(a \cup b)^*$$

مثال:

اوجد اللغة المقبولة للالة التالية:

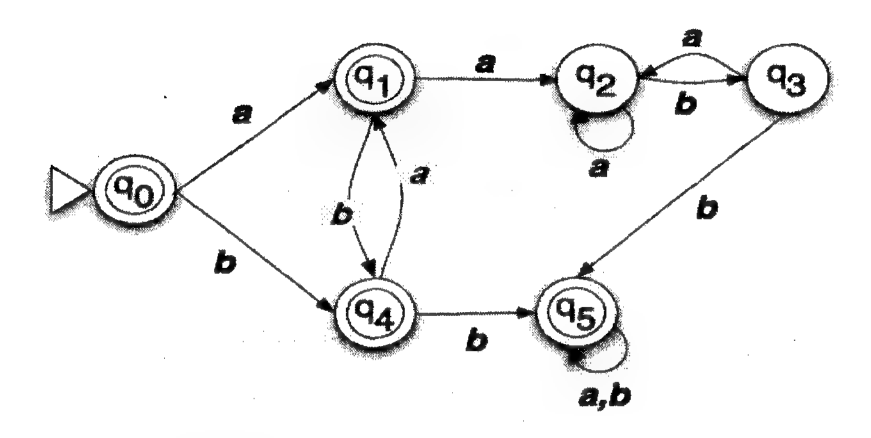


ab^* and $(ab^*aa^*b \cup bb^*a)(aa^*b \cup bb^*a)^*$, so L(M) is

$$ab^* \cup (ab^*a^+b \cup b^+a)(a^+b \cup b^+a)^*$$

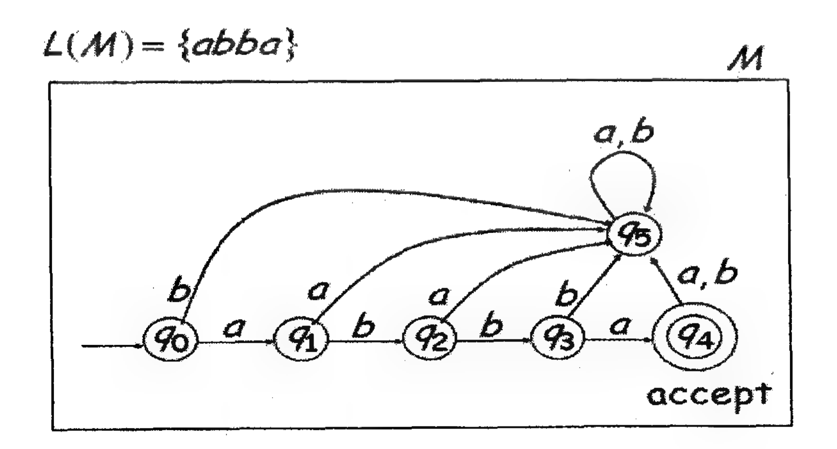
مثال:

اوجد اللغة المقبولة للالة التالية:

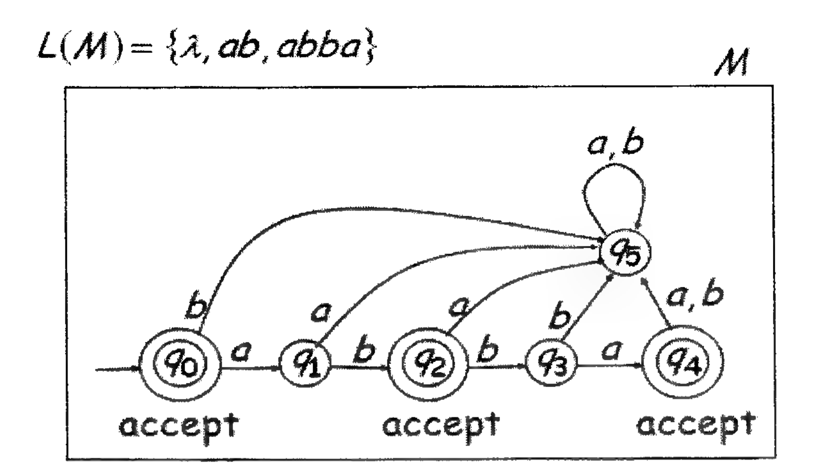


$$\lambda \cup a(ba)^* \cup b(ab)^* \cup (a(ba)^*((a^+b)^+b \cup bb) \cup b(ab)^*(a(a^+b)^+b \cup b))(a \cup b)^*$$

مثال:

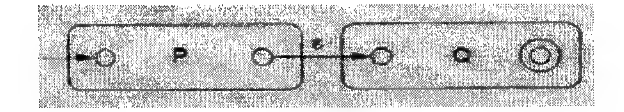


مثال:



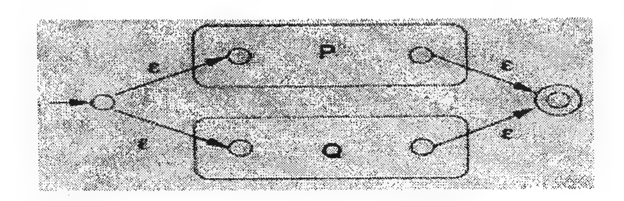
عند استخراج اللغة المقبولة للالة المنتهية لا بد من الانتباه الى الحالات التالية:

- اذا قبلت الة منتهية لغة محددة وكانت متبوعة بالة منتهية اخرى فان اللغة الناتجة هي حاصل دمج اللغتين وكما هو مبين في الشكل التالي:



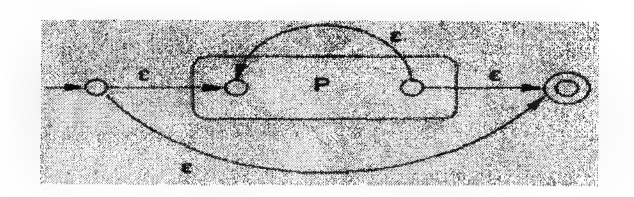
PQ

- اذا كان هناك تفرع لالتين فان اللغة الناتجة هي لغة الالة الاولى اوالثانية:



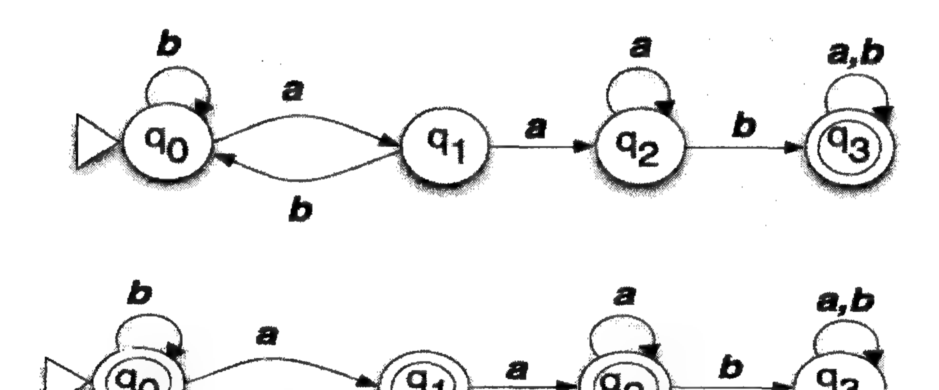
P+Q

- اذا كان هناك حلقة تشكل رجوع داخل الالة فان اللغة المقبولة هي تكرار لمجموعة الرموز P*.



5.2 اللغة المكملة Complement Language

الالة المنتهية المكملة لالة منتهية اخرى هي نفس هذه الالة ولكن بعكس الحالات والمثال التالي يبين الة منتهية تقبل الاحرف aab ومكملها والذي لا يقبل مجموعة هذه الاحرف:



يمثل مكمل اللغة لالة الحالة المتهية مجموعة الرموز المرفوضة من قبل الالة:

Language accepted by M :

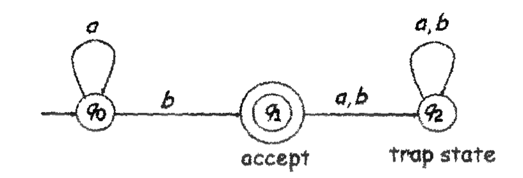
$$L(M) = \{w \in \Sigma^* : \delta^*(q_0, w) \in F\}$$

Language rejected by M :

$$\overline{L(M)} = \{w \in \Sigma^* : \delta^*(q_0, w) \notin F\}$$

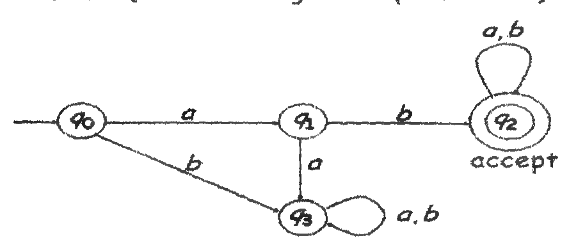
مثال:

$L(M) = \{a^n b : n \geq 0\}$



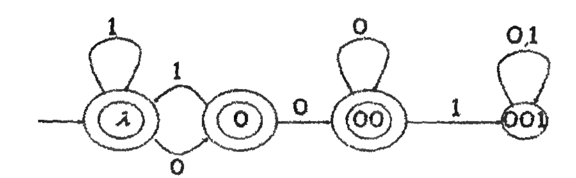
مثال:

$L(M)$ = { all substrings with prefix ab }



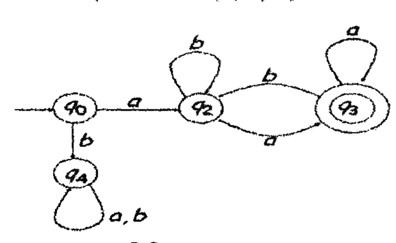
مثال:

$L(M)$ = { all strings without substring 001 }



مثال:

$L = \{awa : w \in \{a,b\}^*\}$

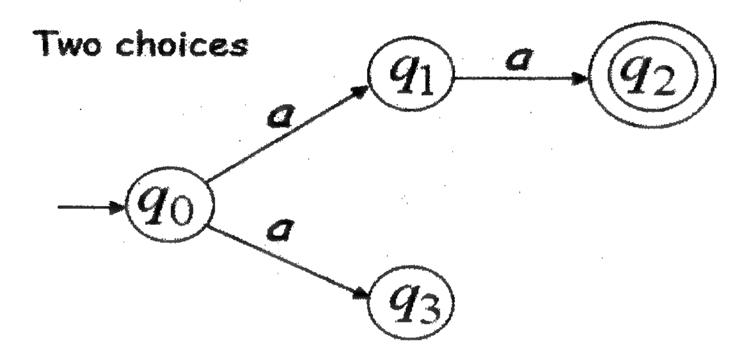



الوحدة الثالثة
آلة الحالة المنتهية
غير المحدودة
Non- Deterministic
Finite State Machine(NDFSM)
3

1.3 تعريف الة الحالة المنتهية غيرالمحدودة:

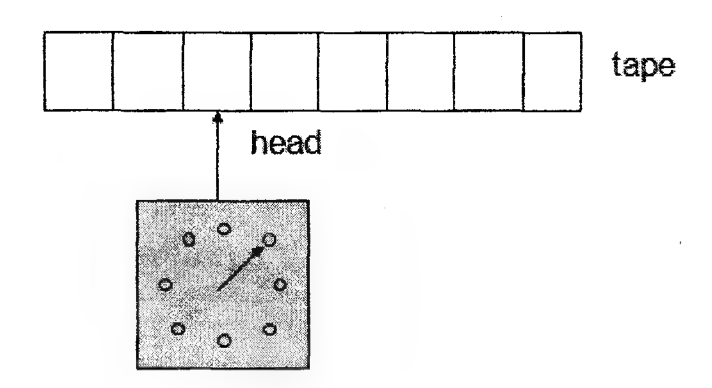
تشبه الة الحالة المنتهيه غير المحدودة الالة المحدودة والتي تعرضنا اليها في الوحدة السابقة ولكن بخلاف بسيط الا وهو امكانية النتقال الى اكثر من حالة عند قراءة الرمز المحدد والمثال التالي يبين نموذجا لالة منتهية غير محدودة

Alphabet = $\{a\}$



لاحظ امكانية الانتقال من الحالة الابتدائية الى حالتين باستخدام نفس الرمز والذي يدل على وجود خيارين.

تتكون الالة المنتهية غير المحدودة من شريط من المدخلات ووحدة تحكم تحتفظ بالحالات وكما هو مبين في الشكل التالي:

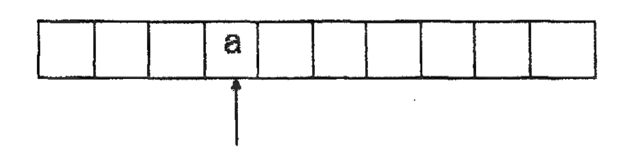


Finite Control

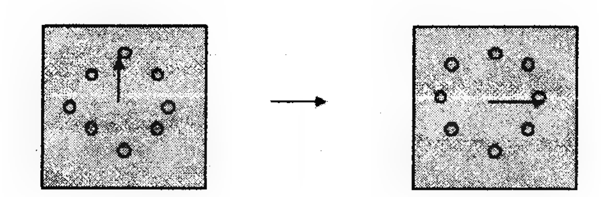
يقسم الشريط الى عدد محدود من الخلايا بحيث يتم تخصيص كل خلية لرمز من الرموز المشكلة لمجموعة الرموز المراد قراءتها من قبل الالة المنتهية غير المحدودة:

a	l	p	h	a	b	e	t

يستخدم راس القراءة لقراءة الرمز من الشريط ونقل الراس الى اليمين ولخطوة واحدة بعد قراءة الرمز او يمكن ابقاء الراس في مكانه في حالة قراءة لامدا اوايبسلون.

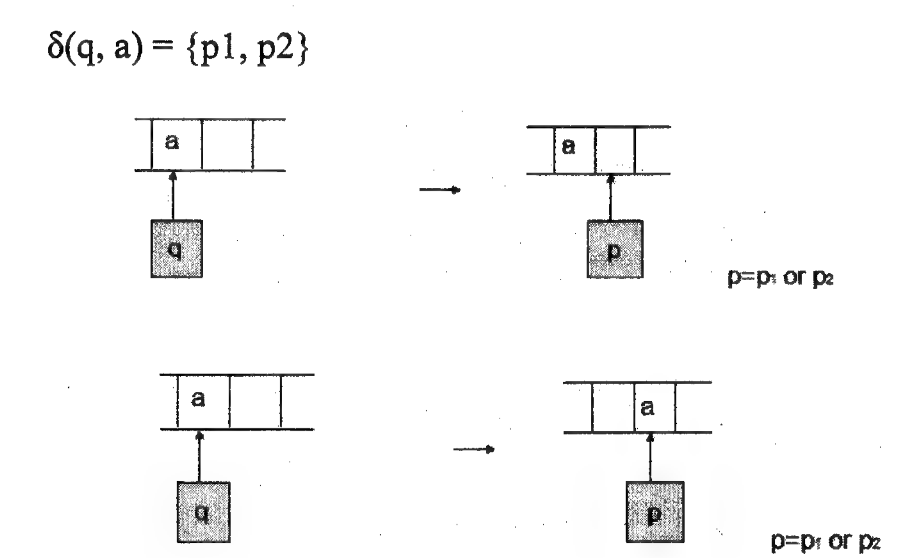


تنتقل الالة المنتهية غيرالمحدودة من حالة الى اخرى بعد قراءة الرمز ويمكن ان تبقى في نفس الحالة.

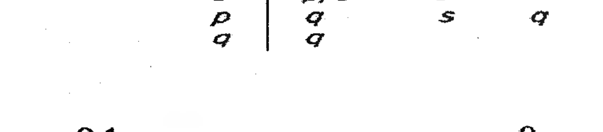


يمكن وصف الالة المنتهية غير المحدودة باستخدام مخطط الحالات واستخدام جدول الانتقال ودالة الانتقال والتي تحدد الحالة المقبلة عند قراءة الرمز ووجود الالة في حالة محددة مثل:

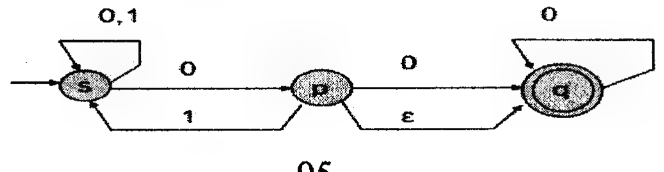
$\delta(q, a) = \{p1, p2\}$



والمثال التالي يبين جدول الانتقال ومخطط الحالات لالة منهية غير محدودة:



δ	0	1	ε
s	p, s	s	
p	q	s	q
q	q		



يتم وصف الالة المنتهية غير المحدودة والتعبير عنها رياضيا بخمسة امور هي:

- مجموعة الحالات.
- مجموعة رموز المدخلات.
- دالة الانتقال.
- الحالة الابتدائية والتي تنتمي الى مجموعة الحالات.
- الحالة(اواكثر) النهائية والتي تنتمي هي الاخرى لمجموعة الحالات:

$$M = (Q, \Sigma, \delta, q_0, F)$$

Q : set of states

Σ : input alphabet

δ : transition function

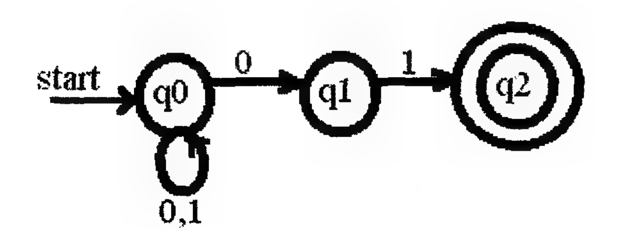
q_0 : initial state

F : set of accepting states

وفيما يلي نستعرض بعض الامثلة:

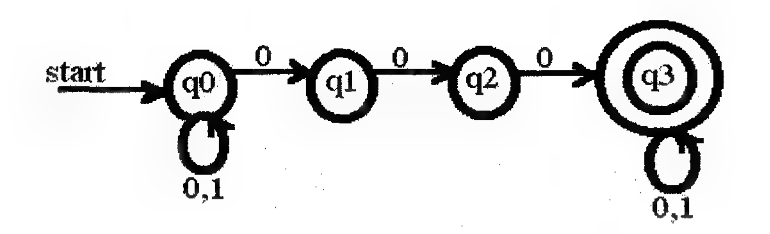
مثال

ابن الة الحالة المنتهية غيرالمحدودة والذي يقبل مجموعة من الاصفار والوحدات والمنتهية بصفر وواحد:

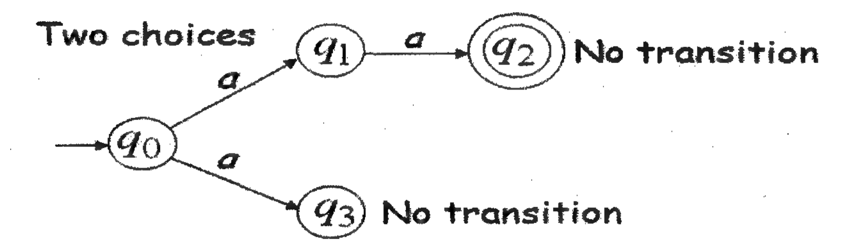


مثال:

ابن الة الحالة المنتهية غيرالمحدودة لقبول سلسلة تحتوي على 3 اصفار متتابعة:

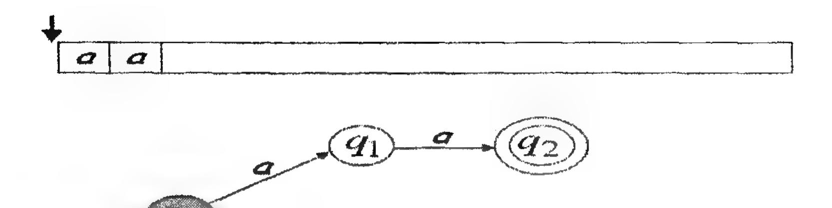


كما اشرنا سابقا فان سابقا فان الالة المنتهية غيرالمحدودة يتوفر فيها اكثر من خيار للتنقل عند قراءة الرمز بحيث يؤدي السير في الخيار الاول الى النتهاء دون الانتقال الى الخيار الثاني وكذلك الحال فيما لو سلكت الالة الخيار الثاني اولا وكما هو مبين في الشكل التالي:



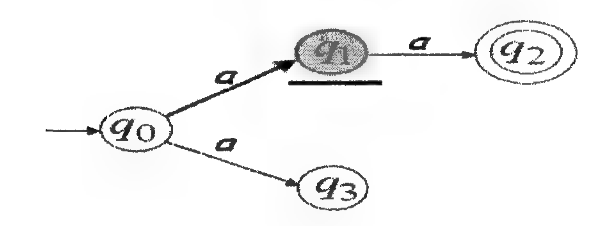
وفيما لو سلكت الالة المسار الاول فان تتابع التنفيذ سيكون كما يلي:

.1



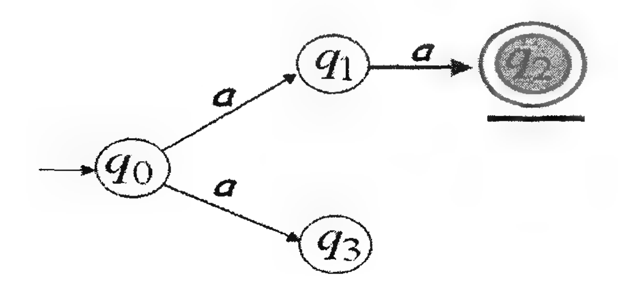
.2





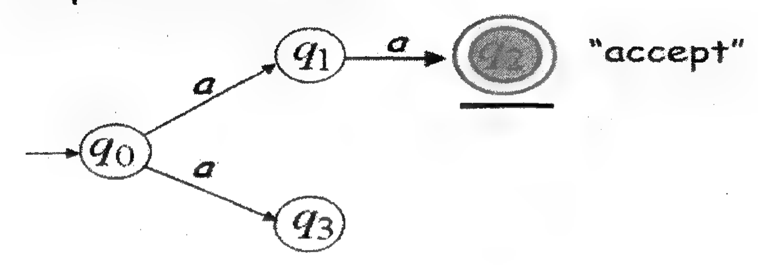
.3





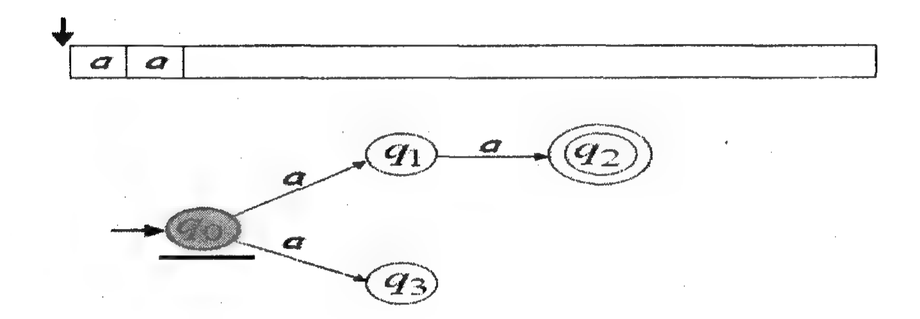
.4





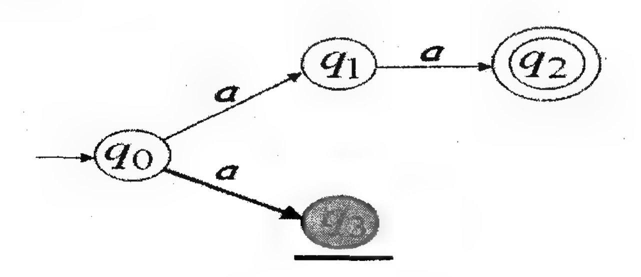
اما اذا سلكت الالة المسار الثاني فان الية التنفيذ ستكون كما يلي:

.1

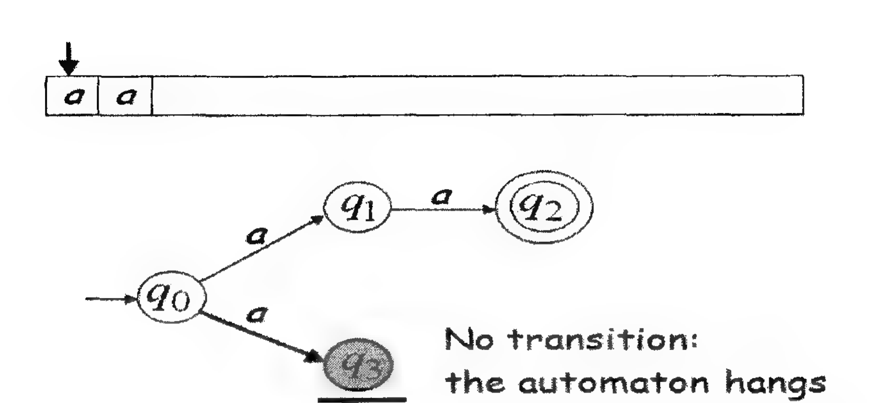


.2





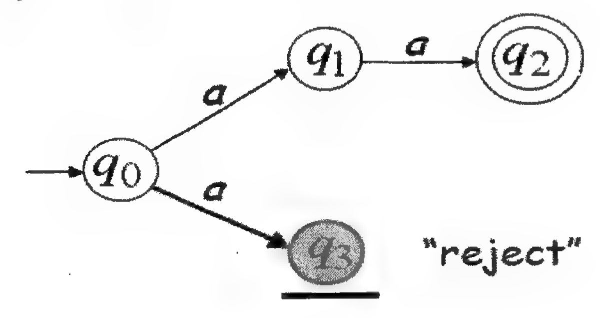
.3



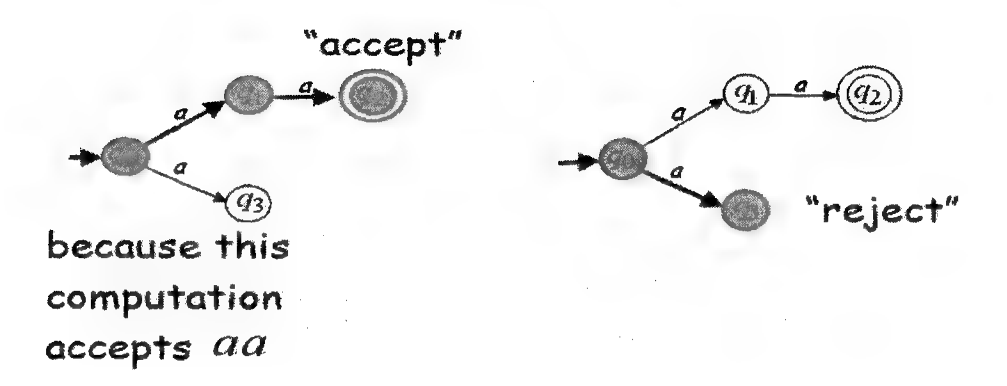
.4



Input cannot be consumed

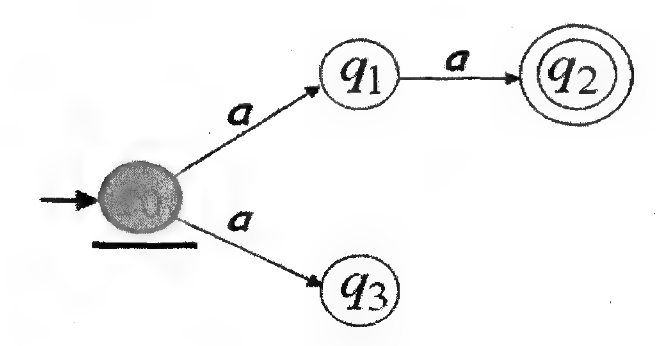


وعليه فان الالة السابقة تقبل اللغة aa وترفض a وكما هو مبين في الشكل التالي:



مما تقدم يتبين ان الالة ترفض سلسلة معينة من الرموز اذا كانت نتيجة الالة بعد قراءة هذه السلسلة غيرمحسوبة اي اذا انتهت عملية قراءة الرموز دون الوصول الى حالة نهائية فمثلا الالة التالية ترقض السلسلة a وفي المسارين:

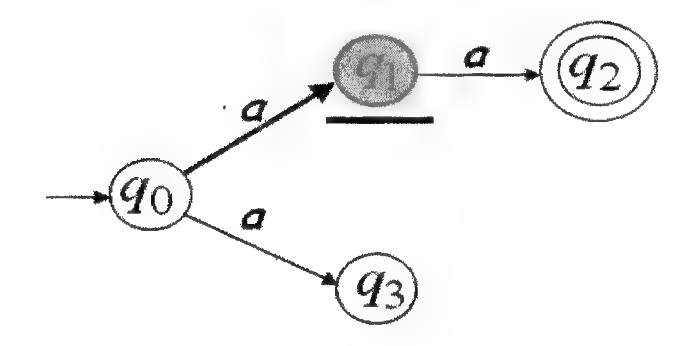




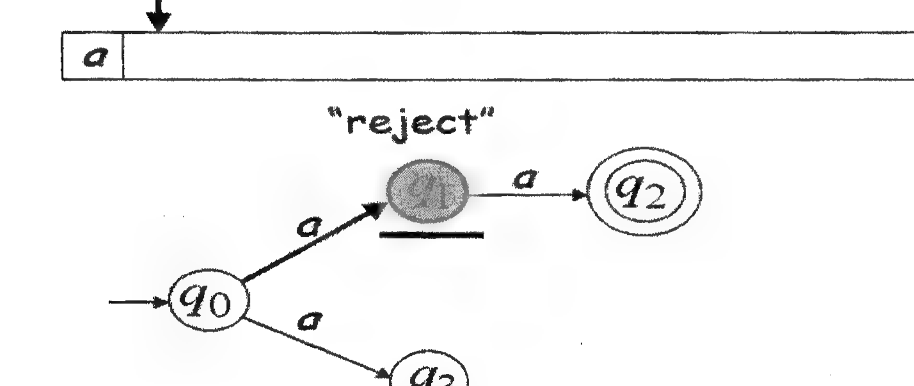
المسار الاول:

.1





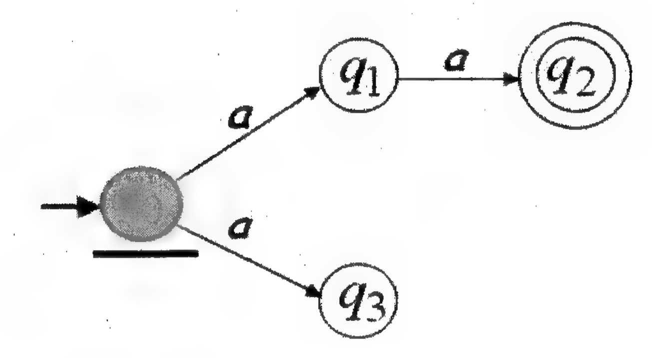
2



المسار الثاني:

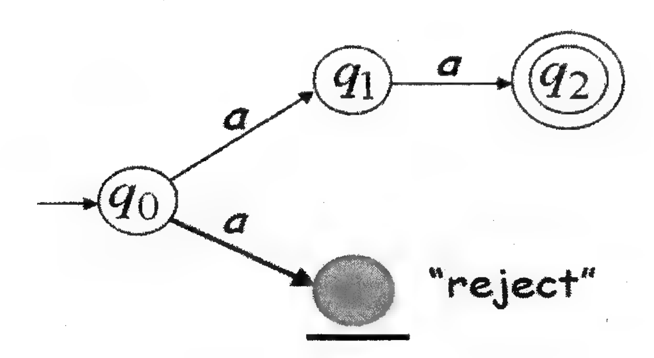
.1





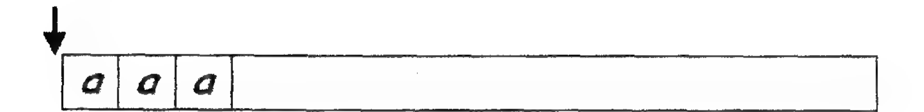
.2

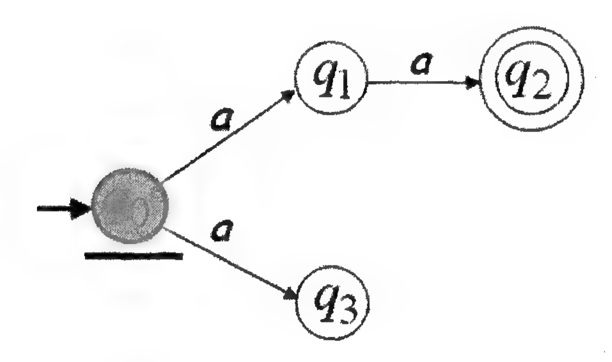




تعتبر الة الحالة المنتهية غيرالمحدودة غيرمحسوبة في الحالات التالية:

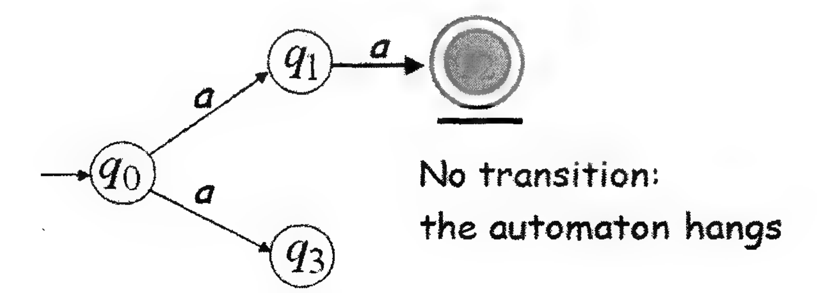
1. اذا انتهت مجموعة الرموز دون الوصول الى حالة نهائية.
2. اذا ادت عملية قراءة الرموز الى الوصول الى حالة غيرنهائية.
3. اذا تم الوصول الى حالة نهائية قبل انهاء اللغة المطلوبة فان السلسلة الرمزية المؤلفة للغة سيتم رفضها وما هو مبين في المثال التالي:

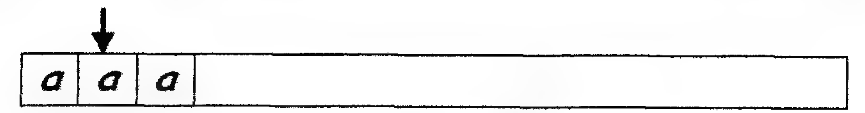




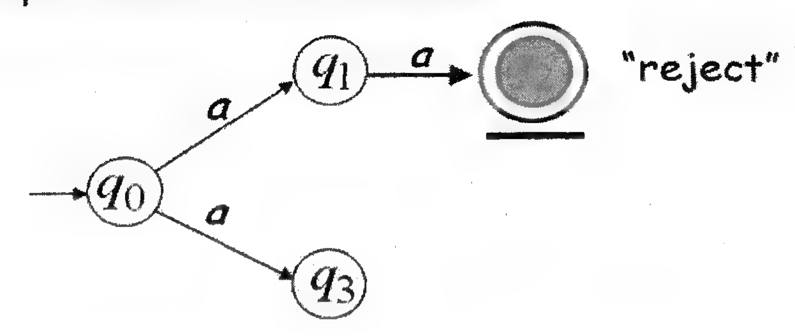
اذا سلكت الالة المسار الاول فان النتيجة النهائية ستكون رفض السلسلة وكما هو مبين ادناه:



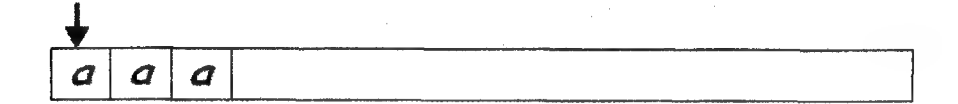


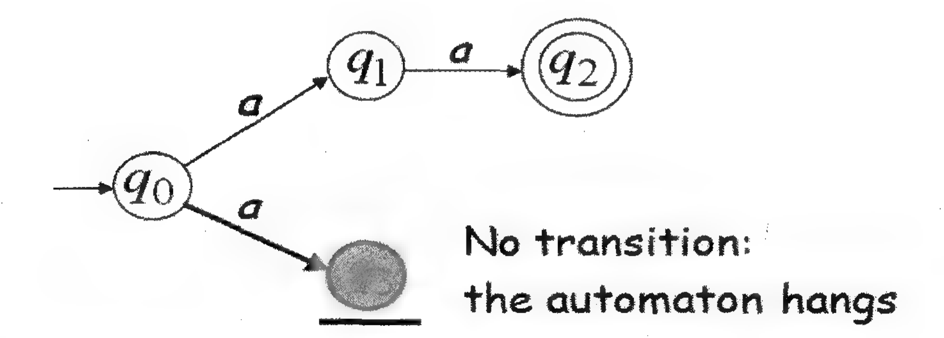


Input cannot be consumed

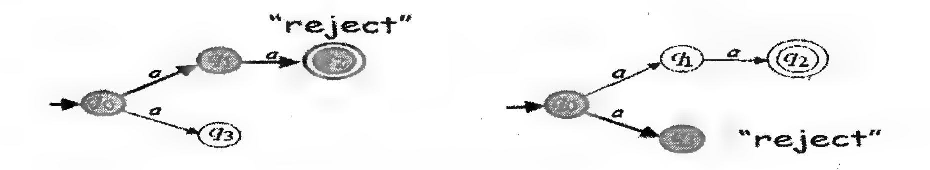


اما اذا سلكت الالة المسار الثاني فان النتيجة ستكون رفض السلسلة للوصول الى حالة غيرمتفرعة وليست نهائية قبل اكمال السلسلة





aaa is rejected by the NFA:



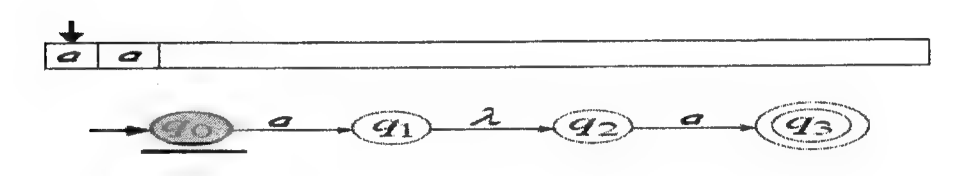
All possible computations lead to rejection

يمكن التخلص من بعض مشاكل رفض اللغة في الالة المنتهية غير المحدودة وتحويلها من الة غيرمحسوبة الى الة محسوبة وذلك باستخدام لامدا أو ايبسلون لتمثيل عملية الانتقال من حالة الى اخرى حيث يمثل هذا الرمز رمزا فارغا وعند المرور به فان راس القراءة يبقى ثابتا على الرمز الذي كان واقفا عليه قبل الانتقال الى الحالة الجديدة باستخدام لامدا.

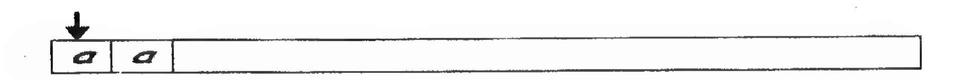
وفيما يلي مثالا يوضح كيفية استخدام لامدا لجعل الالة المنتهية محسوبة:



.1



.2

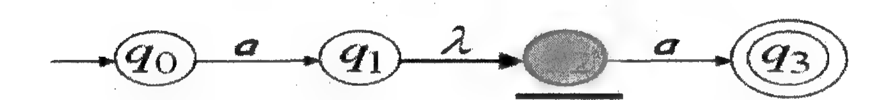




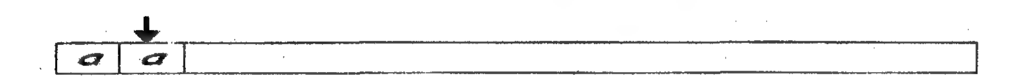
.3

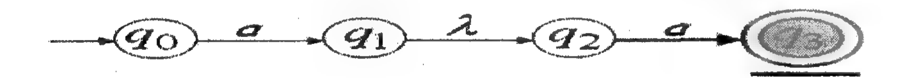
(read head does not move)





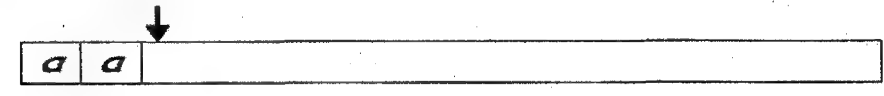
.4

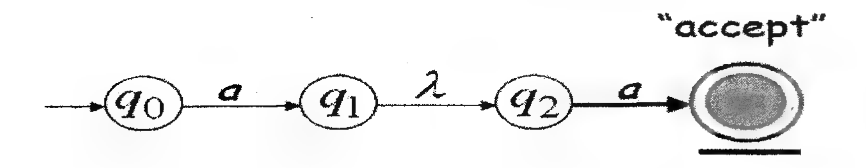




.5

all input is consumed

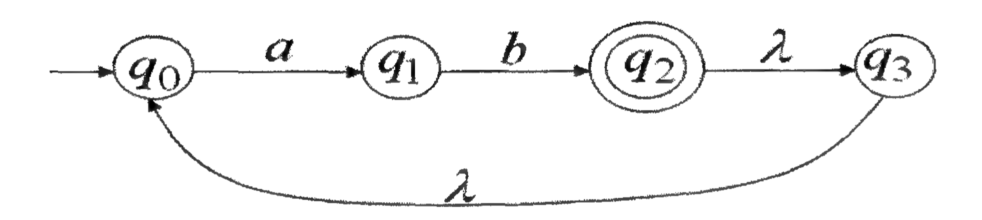




String aa is accepted

هـذا ويمكـن تكـرار مجموعـة الرمـوز المشكلة للغـة المقبولـة مـن قبـل الالـة باستخدام لامدا وكما هو مبين في المثال التالي:

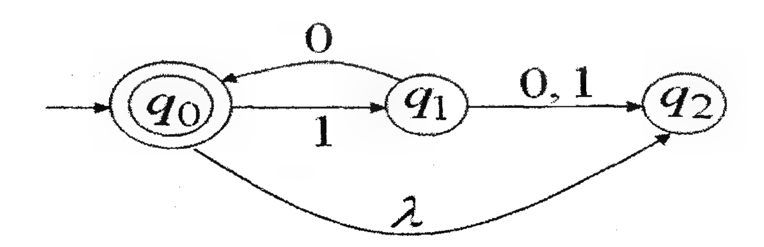
$$L = \{ab,\ abab,\ ababab,\ \ldots\}$$
$$= \{ab\}^+$$



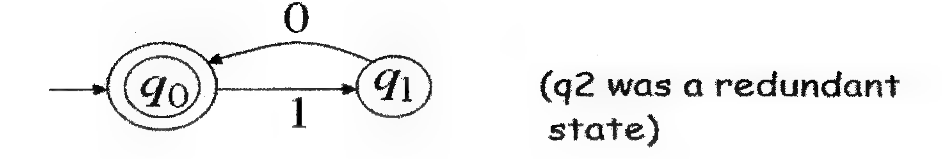
2.3 لغة الالة المنتهية غير المحدودة:

كما اشرنا سابقا فان اللغة المقبولة من قبل الالة المنتهية غير المحدودة هي مجموعـة الرمـوز الـتي لـو قـرات فانهـا تقـود الى حالـة نهائيـة أوتـصبح بعـدها الالـة محـسوبة والـشكل التـالي يـبين مخطـط لالـة منتهيـة واللغـة المقبولـة مـن قبـل هـذه الالة:

$$L(M) = \{\lambda,\ 10,\ 1010,\ 101010,\ \ldots\}$$
$$= \{10\}^*$$



لاحظ في المثال السابق ان الحالة الثانية هي فائضة وعليه يمكن الاستغناء عنها لتحقيق أوتنفيذ نفس اللغة.

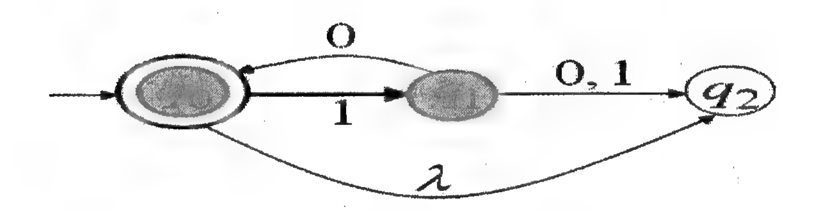


هذا ويمكن استخدام دالة النقل لتحيد لغة الالة المنتهية غيرالمحدودة.

تعبر دالة النقل عن الحالة التي يمكن النتقال اليها بمعرفة الحالة الحالية والرمز المراد قراءته والامثلة التالية تبين بعض دوال الانتقال لالة منتهية غير محدودة:

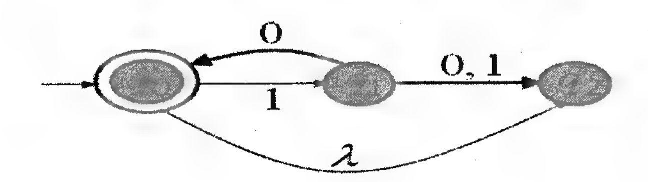
.1

$$\delta(q_0, 1) = \{q_1\}$$



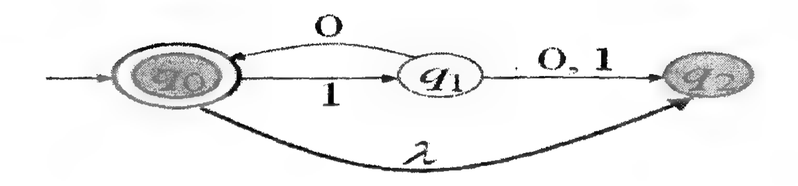
.2

$$\delta(q_1, 0) = \{q_0, q_2\}$$



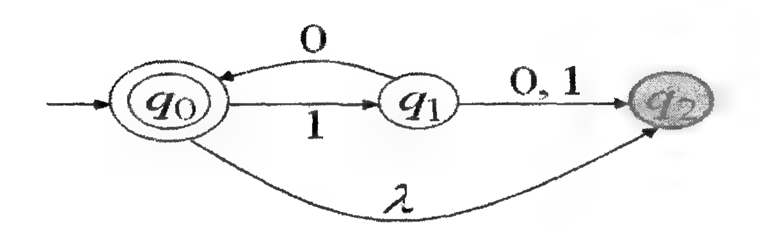
.3

$$\delta(q_0, \lambda) = \{q_0, q_2\}$$



.4

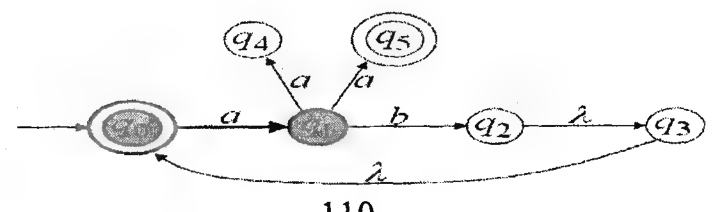
$$\delta(q_2, 1) = \varnothing$$



هذا ويستخدم مفهوم دالة الانتقال الموسعة والتي تشير الى الحالة أوالحالات التي يمكن الانتقال اليها من حالة حالية بعد قراءة سلسلة من الرموز والامثلة التالية تبين أمثلة على دالة الانتقال الموسعة:

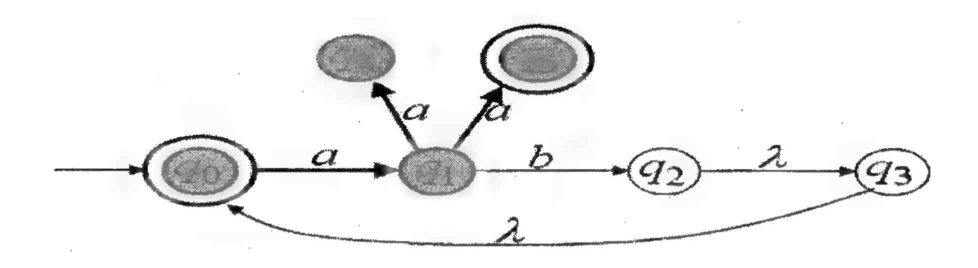
.1

$$\delta^*(q_0, a) = \{q_1\}$$



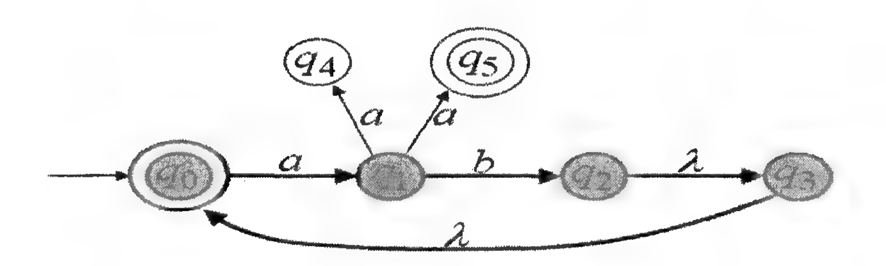
.2

$$\delta^*(q_0, aa) = \{q_4, q_5\}$$



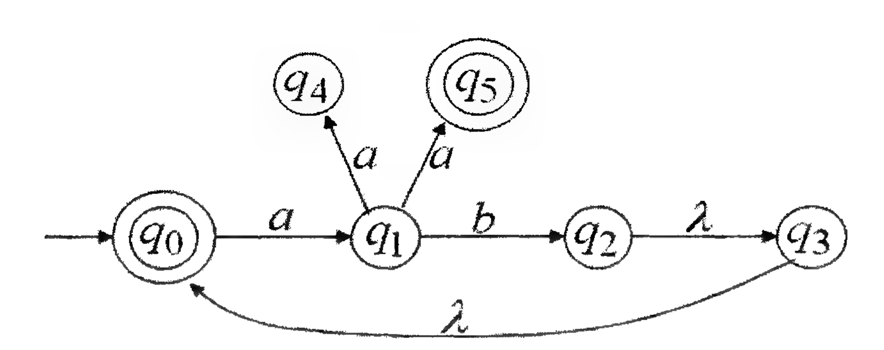
.3

$$\delta^*(q_0, ab) = \{q_2, q_3, q_0\}$$



ولتحديد اللغة المقبولة من الالة المنتهية غير المحدودة يمكن استخدام دوال الانتقال الموسعة فاذا كان الطرف الايمن أو احد عناصرة يشكل حالة نهائية في مجموعة الرموز الواقعة في الطرف الايسر من الدالة تشكل جزءا من اللغة المقبولة كونها تقود الى حالة نهائية.

لناخذ المثال التالي:



نحدد الحالات النهائية ودوال الانتقال الموسهة الى كل حالة من هذه الحالات:

.1

$F = \{q_0, q_5\}$

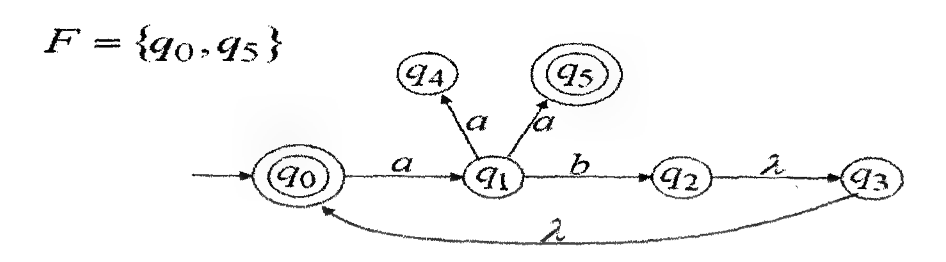
$\delta^*(q_0, aa) = \{q_4, \underline{q_5}\}$ $\searrow \in F$ $\qquad aa \in L(M)$

.2

$F = \{q_0, q_5\}$

$\delta^*(q_0, ab) = \{q_2, q_3, \underline{q_0}\}$ $\searrow \in F$ $\qquad ab \in L(M)$

.3



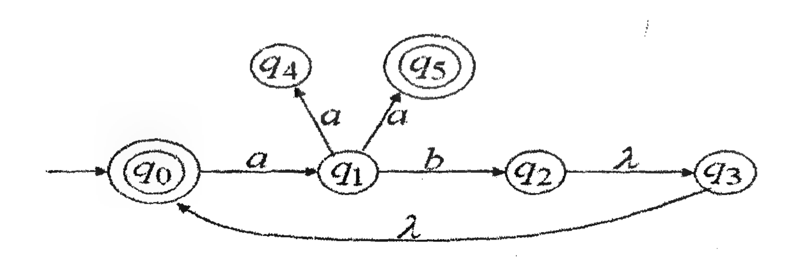
$\delta^*(q_0, abaa) = \{q_4, \underline{q_5}\} \searrow \in F \qquad aaba \in L(M)$

.4

$F = \{q_0, q_5\}$

$\delta^*(q_0, aba) = \{q_1\} \searrow \notin F \qquad aba \notin L(M)$

.5



$L(M) = \{\lambda\} \cup \{ab\}^* \{aa\}$

3.3تحويل الة الحالة المنتهية غيرالمحدودة الى الة محدودة:

اشرنا سابقا الى ان الالة غيرالمحدودة تختلف عن الالة المحدودة في احتواء الالة غيرالمحدودة على اكثر من مسار عند تنفيذ لغة مؤلفة من مجموعة من الرموز المحددة وفيما يلي نبين اوجه الخلاف بين هاتين الالتين وبشكل اكثر تفصيلا:

1. في الالة المحدودة تكون كل عملية انتقال محددة اما في الالة غيرالمنتهية فيمكن ان يكون لهيئة الالة (q, a) اكثر من مسار للانتقال الى حالة جديدة و باستخدام نفس الرمز(بدون انتقال،مسار أومساران أواكثر).
2. في الالة المنتهية تقود عملية قراءة محموعة الرموز المحددة الى السلوك في مسار واحد محدد اما في الالة غيرالمنتهية فيمكن ان تسلك الالة اكثر من مسار بعد قراءة مجموعة الرموز المحددة.
3. لا تحتوي الالة المنتهية على مسارات بالرمز ايبسلون او لامدا في يمكن ان تحتوي الالة غيرالمنتهية على مسارات من هذا النوع.
4. تختلف دالة الانتقال في الالة غيرالمحدودة عنها في الالة المحدودة وذلك باستخدام الرمز ايبسلون وعليه يمكن بيان اوجه الخلاف هذه رياضيا كما يلي:

- DFA

$M = (K, \Sigma, \delta, s, F)$,
$K: \{q_0, q_1, \ldots, q_{n-1}\}$, *finite set of states*,
Σ: *alphabet*,
s: *start state*,
F: *set of final states*,
$\delta: K\ X\ \Sigma \rightarrow K$, *transition function*

- NFA

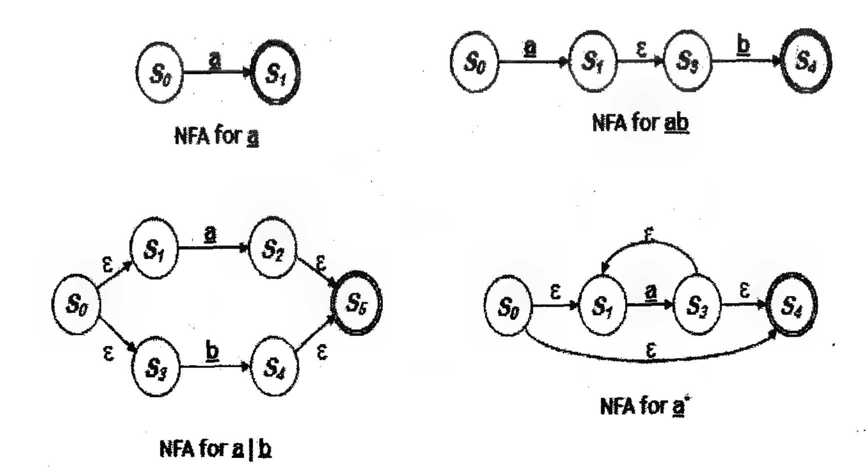
$M = (K, \Sigma, \Delta, s, F)$,
$K: \{q_0, q_1, \ldots, q_{n-1}\}$, *finite set of states*,
Σ: *alphabet*,
s: *start state*,
F: *set of final states*,
$\Delta: K\ X\ (\Sigma U\{\varepsilon\}) \rightarrow K$, *transition relation*

5. يتم قبول مجموعة الرموز في الالة غيرالمنتهية اذا توفر على الاقل مسار واحد مشكل نتيجة لقراءة هذه الرموز وانتهى المسار بحالة نهائية.
6. اذا رفضت مجموعة الرموز في الالة المنتهية فان هذا يعني عدم توفر مسار ينتهي بحالة نهائية ومشكل نتيجة لقراءة هذه الرموز.
7. بالرغم من تعدد المسارات في الالة المنتهية غيرالمحدودة الا انه يمكن البدء بتصميها اولا وبعد ذلك يمكن ان تحول الى الة منتهية محدودة.

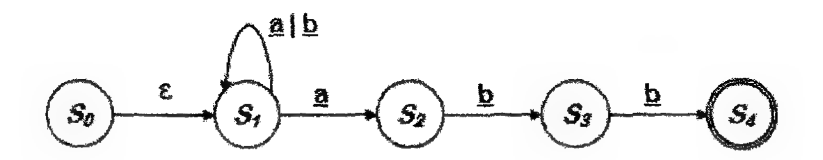
بناء على ما تقدم يمكن القول ان:

1. ايه الة منتهية غيرمحدودة يمكن ان تحول الى الة مكافئة محدودة.
2. اللغة المقبولة من الالة المكافئة يجب ان تكون مطابقة للغة المقبولة من الالة غيرالمحدودة.
3. اذا كان عدد الحالات في الالة غيرالمحدودة مساويا ل س فان عدد الحالات في الالة المكافئة سيكون قريبا من 2 مرفوعة للاس س.

وقبل البدء بعملية التحويل من الالة غيرالمنتهية الى الالة المنتهية لنستعرض بعض الامثلة على التعابير المنتظمة وكيفية بناء الالة غيرالمنتهية لقبولها:



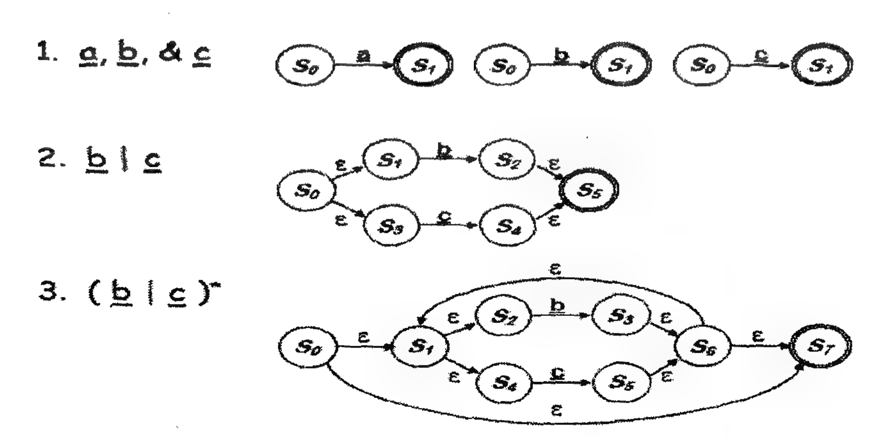
(a | b)* abb



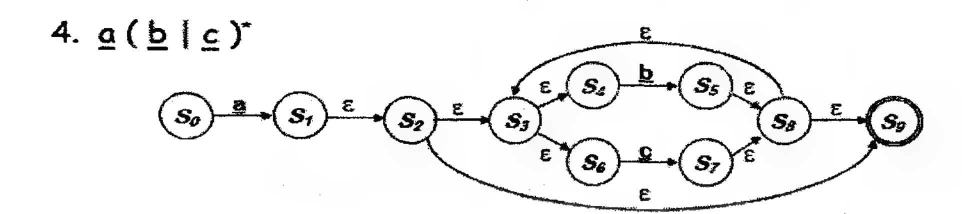
كما اشرنا سابقا فان ايتخدام ايبسلون يؤدي الى تفادي بعض المشاكل في الالة المنتهية غير المحدودة بتحويلها الى الة محسوبة تقرا مجموعة من الرموز للوصول الى حالة نهائية لناخذ المثال التالي والمتضمن نبناء الة منتهية غير محدودة لقبول التعبير المنتظم التالي: a (b | c)*

1. نبدء اولا بكل رمز من الرموز الداخلة في التعبير.
2. الرمز الثاني مربوط مع الرمز الثالث بعلاقة أو اي التفرع باستخدام الرمز الثاني أواستخدام الرمز الثالث.
3. تكرار الخطوة 2 باستخام عملية النتقال بايبسلون.
4. دمج عملية التعرف على الرمز الاول بناتج الخطوة الثالثة.

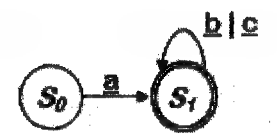
وفيما يلي توضيح لهذه النقاط باستخدام المخطط:



4. a (b | c)*



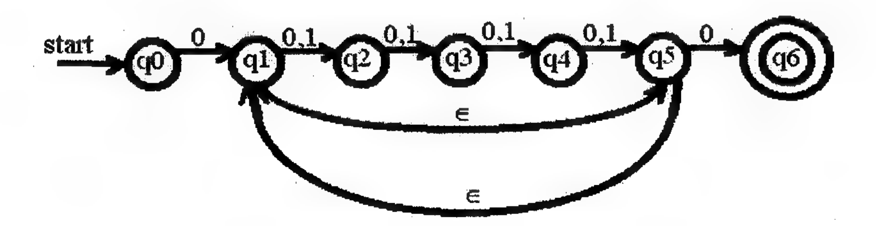
لاحظ انه بدون استخدام ايبسلون يمكن وصف المخطط كما يلي:



يمكن استخام مسار ايبسلون لتنفيذ عملية التكرار لتنفيذ أو قراءة مجموعة من الرموز والاثلة التالية توضح كيفية استخدام مسار ايبسلون لتنفيذ عملية التكرار:

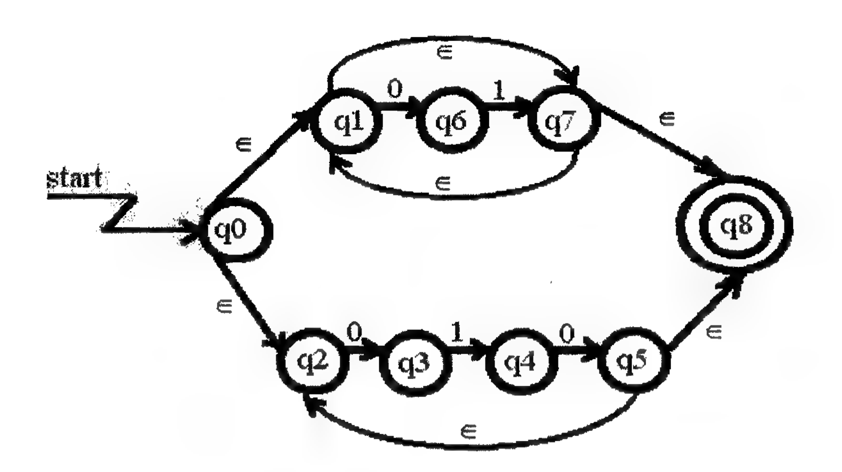
مثال:

ابن الالة المنتهية غيرالمحدودة والتي تقبل اي سلسلة(مكونة من 0 و1) تحتوي عل صفرين مفصولين بسلسلة طولها 4س حيث س اكبر أو تساوي الواحد:



مثال:

اين الة منتهية غيرمحدودة تقبل سلسلة تحتوي على 01 مكررة صفر مرة او اكثر او تحتوي على 010 مكررة مرة واحدة او اكثر:



هناك طرق متعددة للتحويل من الة منتهية غيرمحدودة الى الة منتهية محدودة ومن هذا الطرق استخدام مفهوم التغطية باستخدام ايبسلون والتي تعني ايجاد جميع الحلالات التي يمكن الوصول اليها باستخدام ايبسلون.

تستخدم هذه الطريقة اقترانين مفتاحين هما:

1. مجموعة الحالات التي يمكن الوصول اليها من حالة محددة بعد قراءة رمز او اكثر ويرمز لهذه الدالة ب:

Move (s_i, a)

2. مجموعة حالات التغطية بايبسلون وهي مجموعة الحالات التي يمكن الوصول اليها من حالة محددة باستخدام الرمز ايبسلون ويرمز لها ب:

ε-closure (s_i)

تنفذ خوارزمية التحويل حسب الخطوات التالية:

- Start state derived from s_0 of the NFA
- Take its ε-closure S_0 = ε-closure(s_0)
- Take the image of S_0, Move (S_0, α) for each α ∈ Σ, and take its ε-closure
- Iterate until no more states are added

The algorithm:

s_0 ← ε-closure(q_{0n})

while (S is still changing)
for each s_i ∈ S
for each α ∈ Σ
$s_?$ ← ε-closure(Move(s_i, α))
if ($s_?$ ∉ S) then
add $s_?$ to S as s_j
T[s_i, α] ← s_j

The algorithm halts:

1. *S* contains no duplicates (test before adding)
2. 2^{Qn} is finite
3. while loop adds to *S*, but does not remove from *S (monotone)*

⇒the loop halts

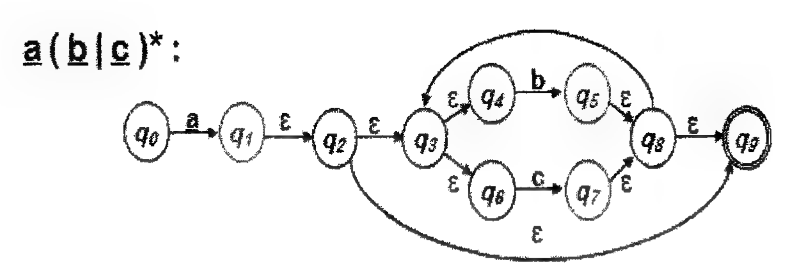
S contains all the reachable NFA states

It tries each character in each s_i.

It builds every possible NFA configuration.

⇒ S and T form the DFA

وفيما يلي نستعرض مثالا نستخدم فيه هذه الخوارزمية لتحويل الالة المنتهية غيرالمحدودة الى الة منتهية محدودة:

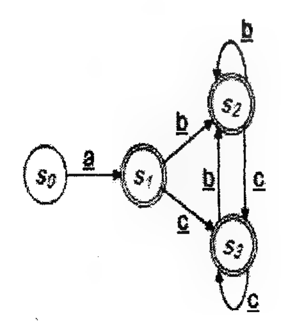


Applying the subset construction:

	NFA states	ε-closure (move(s,*)) a	b	c
s_0	q_0	$q_1, q_2, q_3, q_4, q_8, q_9$	none	none
s_1	$q_1, q_2, q_3, q_4, q_8, q_9$	none	$q_5, q_8, q_9, q_3, q_4, q_6$	$q_7, q_8, q_9, q_3, q_4, q_6$
s_2	$q_5, q_8, q_9, q_3, q_4, q_6$	none	s_2	s_3
s_3	$q_7, q_8, q_9, q_3, q_4, q_6$	none	s_2	s_3

Final states

The DFA for a (b | c)*



δ	a	b	c
s_0	s_1	-	-
s_1	-	s_2	s_3
s_2	-	s_2	s_3
s_3	-	s_2	s_3

4.3 التخلص من ايبسلون في الالة المنتهية غيرالمحدودة:

كما اشرنا سابقا الى فائدة استخدام ايبسولن في الالة المنتهية غيرالمحدودة وذلك لتحويل الالة الى الة محسوبة تصل الى حالة نهائية بقراءة مجموعة من الرموز هذا ويمكن ايضا التخلص من ايبسلون مع المحافظة على اللغة و ابقاء الالة محسوبة ولتنفيذ عملية التخلص من ايبسلون يمكن تنفيذ الخوارزمية التالية والممثلة بمجموعة الخطوات التالية:

1. Compute ε* for the current state, resulting in a set of states S.
2. δ(S,a) is computed for all a in ∑ by
   a. Let S = {p1, p2, ... pk}
   b. Compute I=1→k (pi,a) and call this set {r1, r2, r3... rm}.This set is achieved by following input a, not by following any ε transitions
   c. Add the ε transitions in by computing (S,a)= I=1→ m ε*(r1)
3. Make a state an accepting state if it includes any final states in the -NFA.

وفيما يلي نستعرض مثالا لاستخدام هذه الخوارزمية من اجل التخلص من ايبسلون

لناخذ الالة المنتهية غيرالمحدودة والممثلة كما يلي:

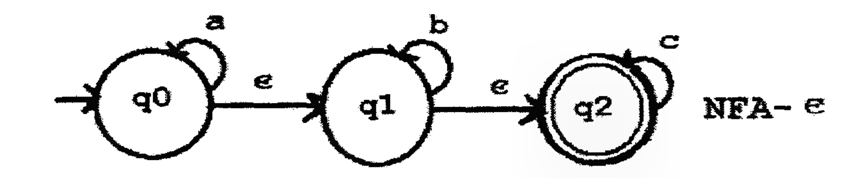
M = {Q,∑,δ, q0, F}

Q = {q0, q1, q2}

∑= {a, b, c} and ε moves

q0 = q0

F = {q2}



دالة الانتقال لهذه الالة هي كما يلي:

δ	a	b	C	ε
q0	{q0}	ϕ	ϕ	{q1}
q1	ϕ	{q2}	ϕ	{q2}
q2	ϕ	ϕ	{q2}	ϕ

نجد الحالات الجديدة كما يلي:

Q' = {{q0, q1, q2}, {q1, q2}, {q2}} or renamed {qx, qy, qz}

Σ= {a, b, c}

F' = {{q0, q1, q2}, {q1, q2}, {q2}} or renamed {qx, qy, qz}

q0 = {q0, q1, q2} or renamed qx

نبني جدول الانتقال باستخدام هذه الحالات:

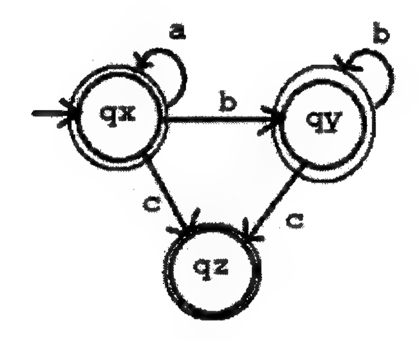
δ'	a	b	c
qx or{q0,q1,q2}			
qy or{q1,q2}			
qz or{q2}			

δ'	a	b	c
qx or{q0,q1,q2}	{qx} or{{q0,q1,q2}}	{qy} or{{q1,q2}}	{qz} or{{q2}}
qy or{q1,q2}	ϕ	{qy} or{{q1,q2}}	{qz} or{{q2}}
qz or{q2}	ϕ	ϕ	{qz} or{{q2}}

δ'	a	b	c
qx	{qx}	{qy}	{qz}
qy	ϕ	{qy}	{qz}
qz	ϕ	ϕ	{qz}

→Q'

وبهذا نحصل على الالة المنتهية غيرالمنتهية وبدون استخدام ايبسلون:



5.3تطبيقات الالات الحالة المنتهية:

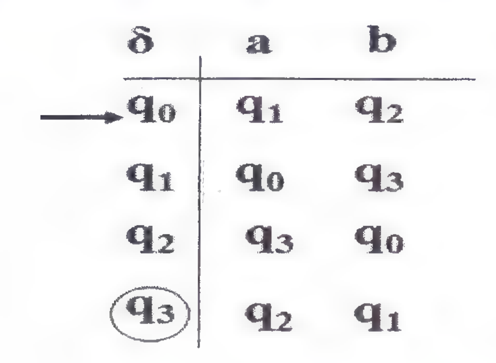
تستخدم الالات المنتهية في كثير من التطبيقات ز من أهم هذه التطبيقات نورد:

1. تمثيل الدارات المنطيقة التتابعية المختلفة وسوف نستعرض نماذح من هذه الالات في نهاية هذا الكتاب ان شاء اللة ونخص بالذكر هنا الة مور والة ميلي.
2. تمثيل وقراءة التعابير المنتظمة واللغات المؤلفة من مجموعة من الرموز.
3. تصميم المترجمات.
4. بناء البرمجيات الخاصة والمعتمدة على عملية تمييز الانماط مثل برامج معالجة الصور والصوت.

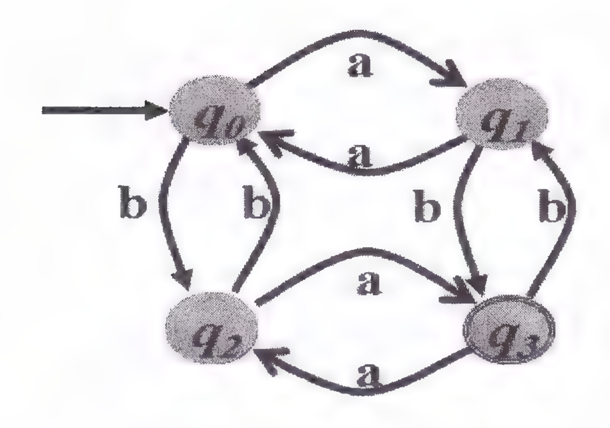
استعرضنا في الوحدة هذه والوحدة السابقة كيفية استخدام الالة لتمثيل مجموعة من الرموز المشكلة لتعبير منتظم اولغة معينة بحيث تعمل الالة المنتهية على قبول هذه المجموعة والتعرف عليها وفيما يلي وللتذكير فقط نورد بعض الامثلة على كيفية تمييز التعابير المنتظمة.

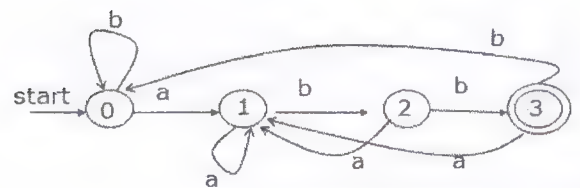
مثال: الالة المنتهية والتي تقبل عددا زوجيا من الاحرف علما بان سلسلة الاحرف مؤلفة من حرفين:

$$\delta(q_0,a) = q_1$$
$$\delta(q_0,b) = q_2$$
$$\delta(q_1,a) = q_0$$
$$\delta(q_1,b) = q_3$$
$$\delta(q_2,a) = q_3$$
$$\delta(q_2,b) = q_0$$
$$\delta(q_3,a) = q_2$$
$$\delta(q_3,b) = q_1$$

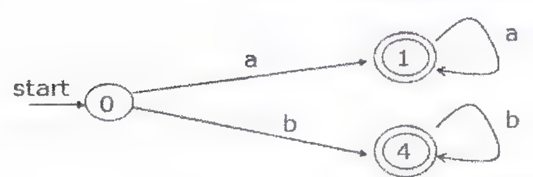


δ	a	b
$\rightarrow q_0$	q_1	q_2
q_1	q_0	q_3
q_2	q_3	q_0
(q_3)	q_2	q_1





Accepted language: (a|b)*abb

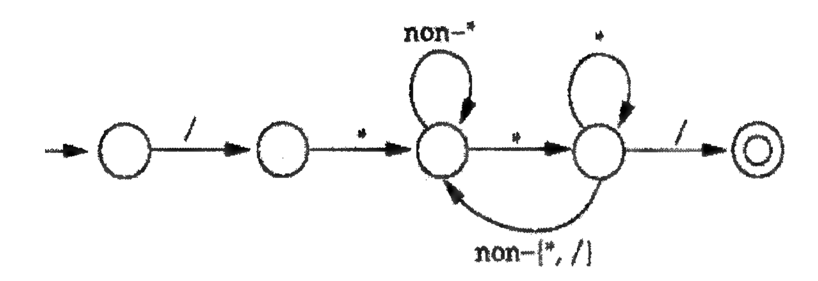


Accepted language: a+ | b+

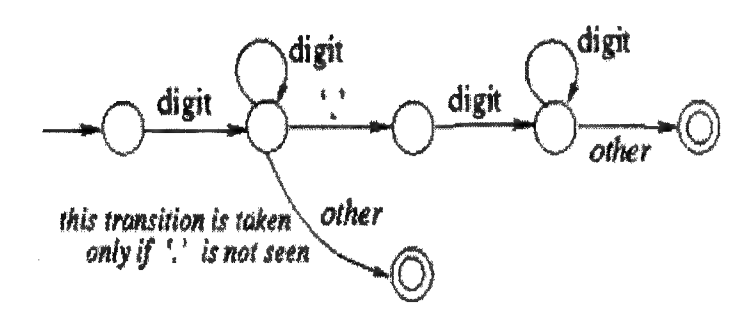
تستخدم الالات المنتهية ايضا في عملية التحليل اللغوي وهي من من المراحل الاساسية لبناء المترجمات والتي تعمل علة ترجمة البرامج المصدرية المكتوبة بلغة كلغة سي بلس باس ويشبه دور الالات النتهية هنا دور الالات المنتهية في

التعرف على التعابير المنتظمة وقبولها وفيما نستعرض بعض مهمات المترجم وكيف يمكن تمثيلها باستخدام الالات المنتهية:

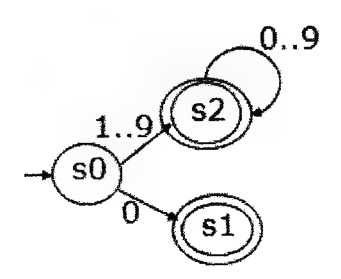
- تمييز الملاحظة في لغة سي بلس بلس والتعرف وقبول هذه الملاحظة اذا كانت بالصيغة المحددة أو رفضها اذا لم تتطابق مع لغة الالة:



- التعرف على القيم الكسرية مثل 123.45



- التعرف على الاعداد الصحيحة.



الوحدة الرابعة

آلة الحالة المنتهية المستخدمة للحزمة

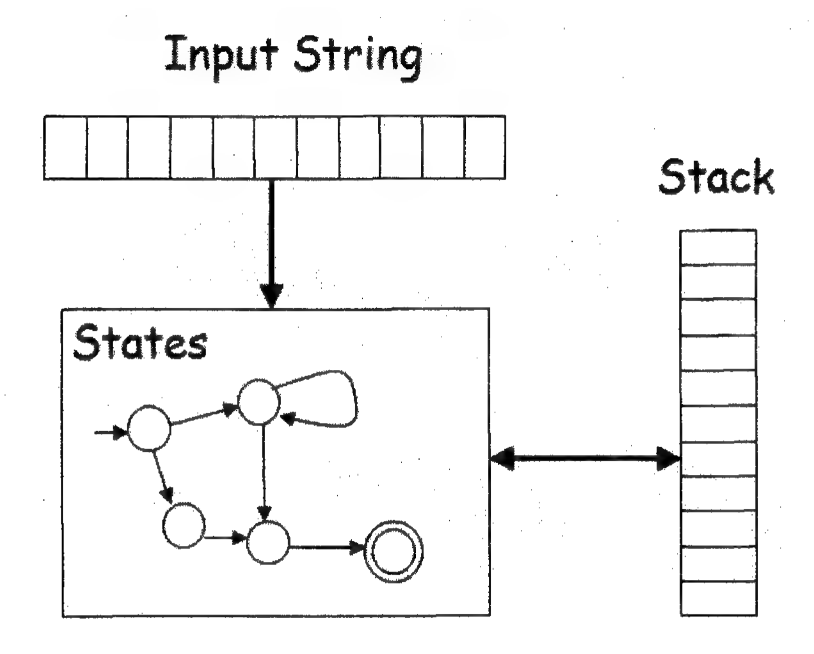
Finite Automata Using Stack

(Push Down Automata)

4

1.4المفهوم العام لالة الحالة المستخدمة للحزمة:

تتكون الالة المنتهية ذات الدف في الحزمة وكما هو مبين في الشكل التالي من شريط المدخلات والذي يحتوي على مجموعة الرموز المراد قراءتها ووحدة تحكم تحتفظ بحالات الة وحزمة تستخدم لعمليت حفظ الرموز واسترجاعها:

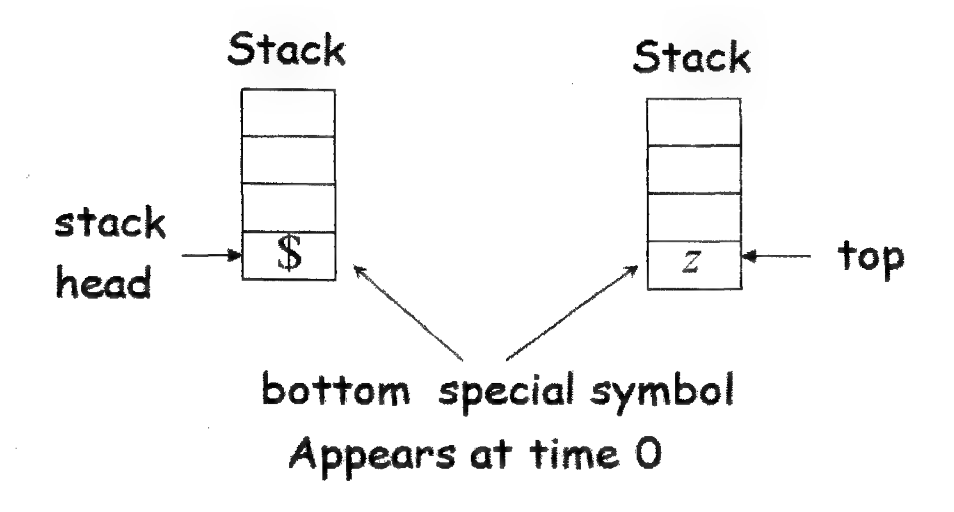


تشبه هذه الالة الى حد كبير الالات المتهية التي استعرضناها في الوحدة الثانية والثالتة والخلاف الوحيد هنا هو استخدام هذه الالة لذاكرة الحزمة والتي تستخدم لحفظ الرموز المقروءة واسترجاعها.

تتالف الحزمة من مجموعة من المواقع وتنفذ عليها عمليتا الدفع(الحفض) في الحزمة والاسترجاع.

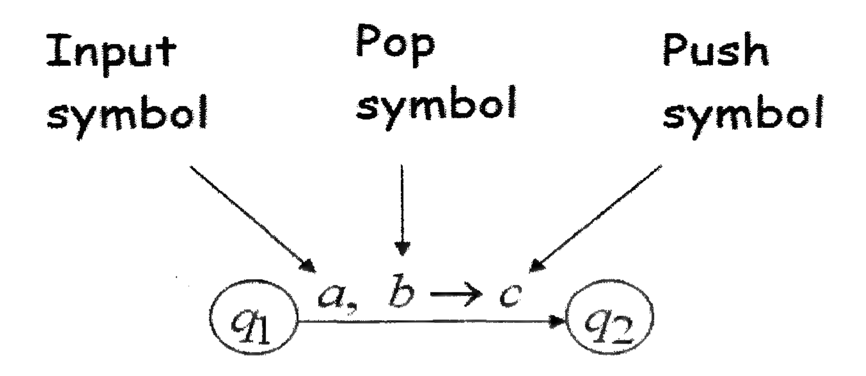
تمتاز الحزمة بخاصية الداخل أولا خارج أولا اي ان اخر رمز حفظ في الحزمة يتم استرجاعه وحذفه من الحزمة أولا.

تبدء الحزمة بحالة ابتدائية يتم وضع رمز خاص فيها للاشارة الى انتهاء الرموز المقروءة وكما هو مبين في الشكل التالي فاذا نفذت الالة المنتهية وقرات مجموعة من الرموز ونتج عن عمليات القراءة تنفيذ عمليات اضافة وحذف من الحزمة واصبحت الحزمة بعد تنفيذ هذه العمليات فارغة فان هذا سيدل الى الوصول الى حالة منتهية أو قبول مجموعة الرموز والتعرف عليها.



كما اشرنا تنفذ عملية الاضافة والحذف على الحزمة من طرف واحد ويستخدم مؤشر يشير الى هذا الطرف وعند تنفيذ عملية الاضافة يزاد المؤشر بقيمة واحد ثم يوضع الرمز في الحزمة وعند الحذف يسترجع العنصر المشار اليه بالمؤشر من الحزمة ويطرح واحد من المؤشر.

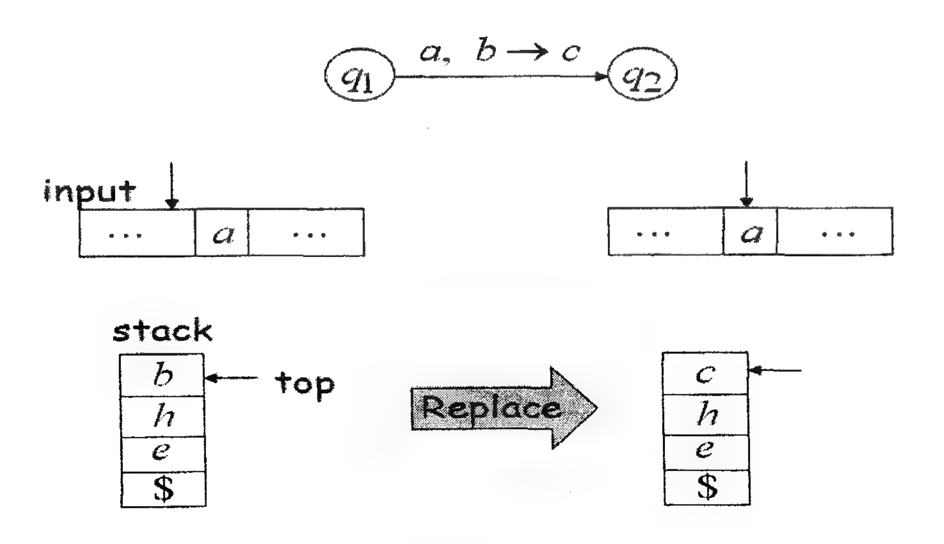
تستخدم الدائرة لتمثيل الحالة والخط المستقيم المنتهي بالسهم يخصص لتمثيل عملية الانتقال من حالة الى اخرى على ان تكتب على الخط نواتج عملية الانتقال وكما هو موضح في الشكل التالي:



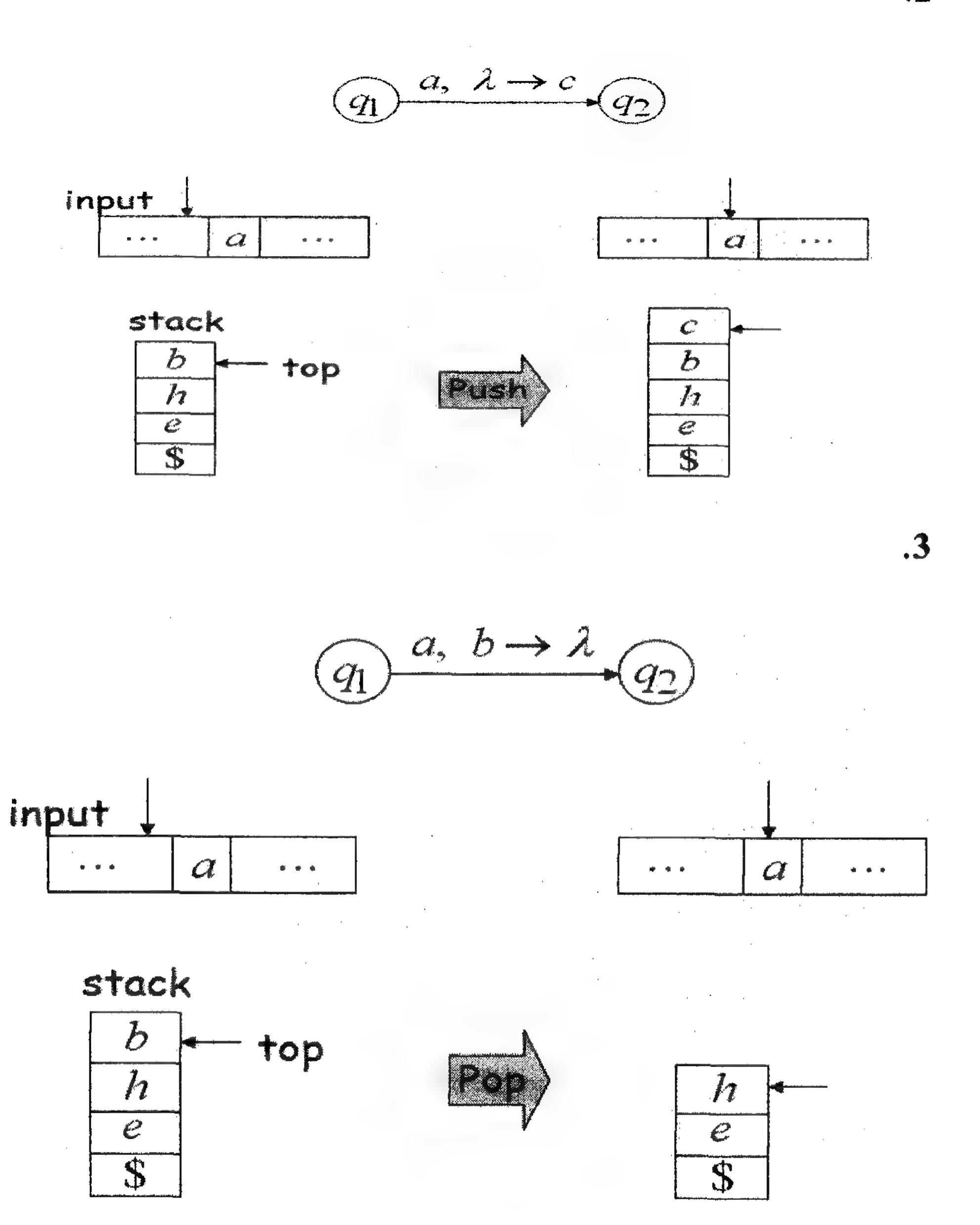
اي ان عملية الانتقال مرتبطة بالحالة الحالية والرمز المراد قراءته و الرمز المراداسترجاعه من الحزمة والرمز المرد حفظه في الحزمة والحالة(اوالحالات اذا كانت الالة غيرمحدودة) المراد الانتقال اليها(بعدالقراءة ينتقل راس القراءة الى اليمين ولموقع واحد على شريط المدخلات).

والاشكال التالية توضح الية تنفيذ عملية الانتقال من حالة الى حالة اخرى وتنفيذ عمليتي الاضافة الى الحزمة والحذف منها.

.1



.2

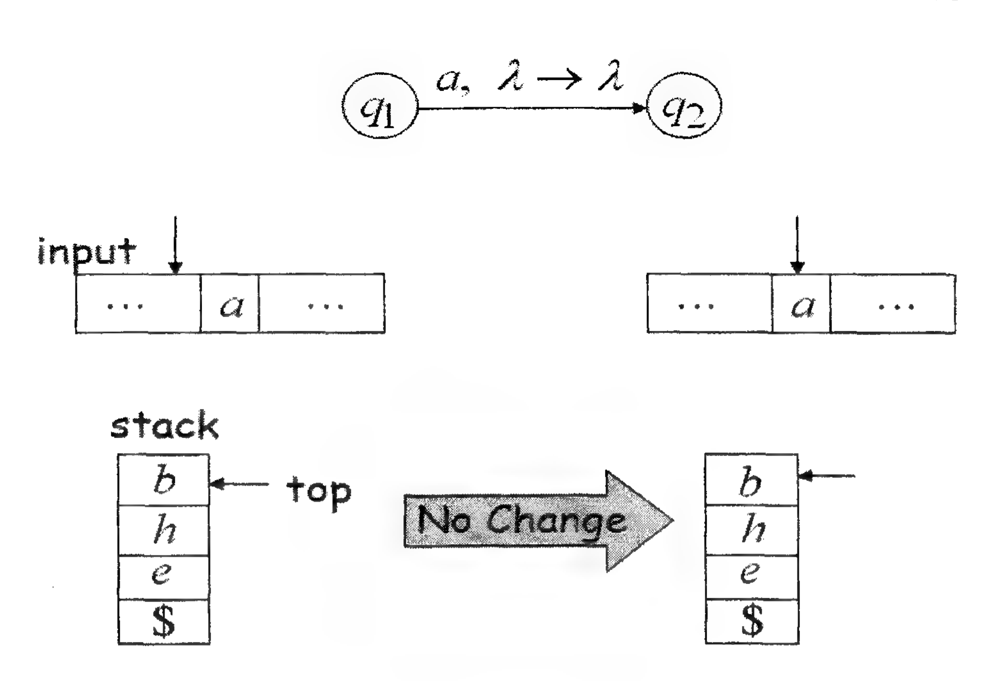


a, λ → c
q1
q2
input
...
a
...
stack
b
h
e
$
top
Push
c
b
h
e
$

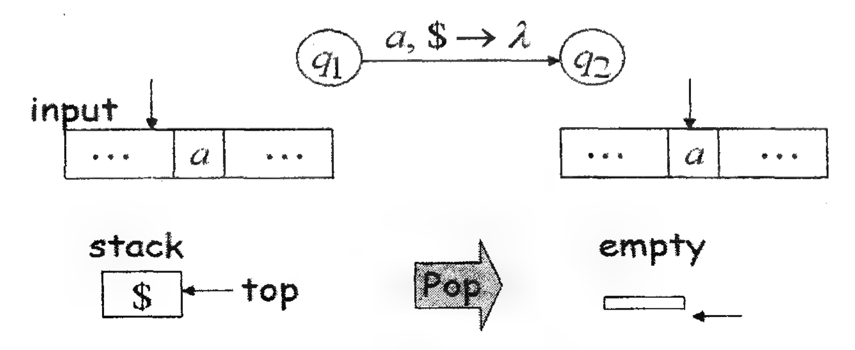
.3

a, b → λ
q1
q2
input
...
a
...
stack
b
h
e
$
top
Pop
h
e
$

.4



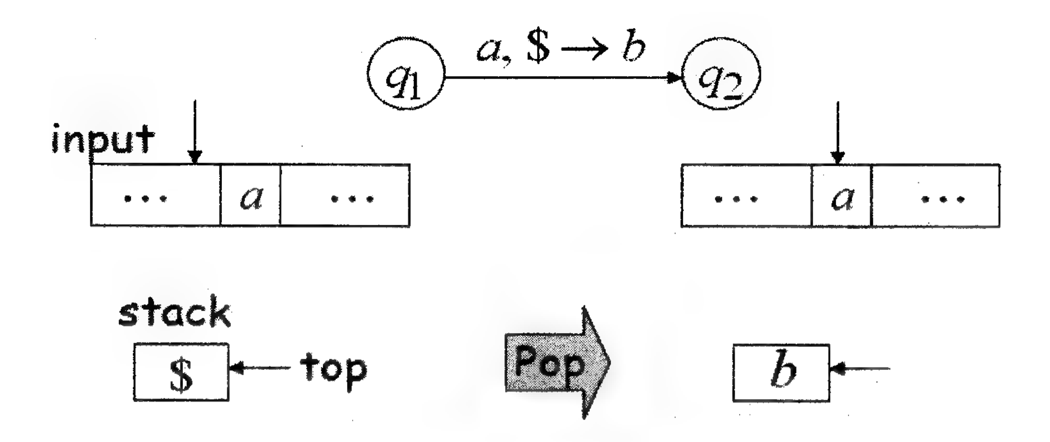
اذا اصبحت الحزمة فارغة فهذا مؤشر الى انتهاء عملية الانتقال وانتهاء الرموز المقروءة فاذا كانت الحالة التي تم الوصول اليها نهائية فان الالة تكون قد ترفت على الرموز وقبلتها والا ستكون مجموعة الرموز المقروءة مرفوضة من قبل هذه الالة والشكل التالي يبين الوصول الى نهاية الرموز بالحصول على حزمة فارغة:



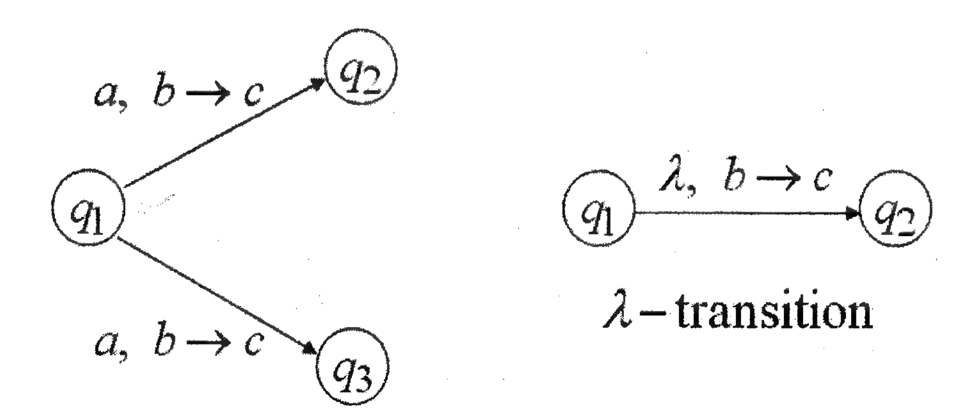
The automaton HALTS

No possible transition after q_2

والشكل التالي يبين عملية انتقال مسموحة.

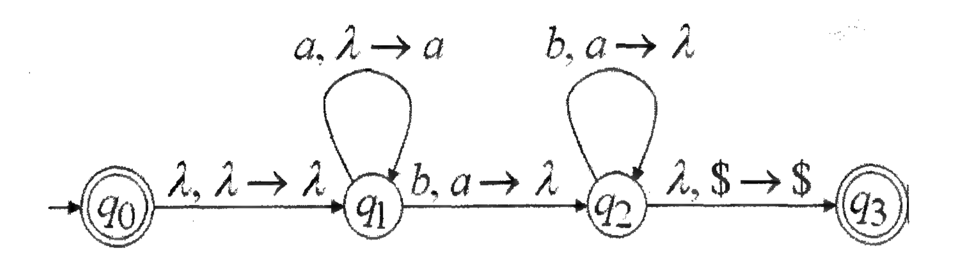


تشبه الالة المنتهية المستخدمة للحزمة الة الحالة المنتهية والتي يمكن ان تكون محدودة أوغير محدودة وعليه فان الالة المنتهية المستخدمة للحزمة يمكن ان تكون محدودة أو غير محدودة ولحل عملية الفرع في المسارات بقراءة نفس الرمز يمكن استخدام عملية الانتقال بلامدا(النتقال بالفراغ او بلا شيئ ودون تحريك راس القراءة تماما كما في الالة المنهية غيرالمحدودة) والشكل التالي يبين هذا:

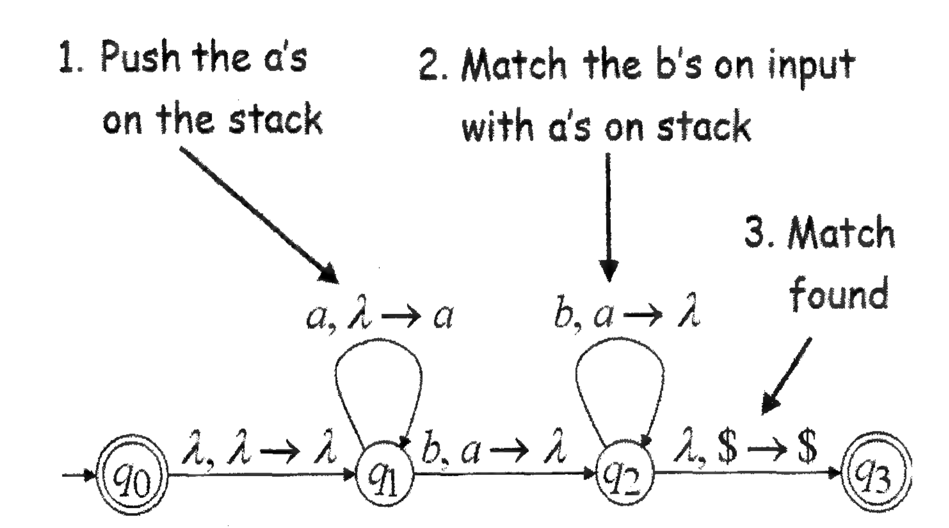


والان لناخذ مثال ونبين كيفية تنفيذ هذه الالة:

$$L(M) = \{a^n b^n : n \geq 0\}$$



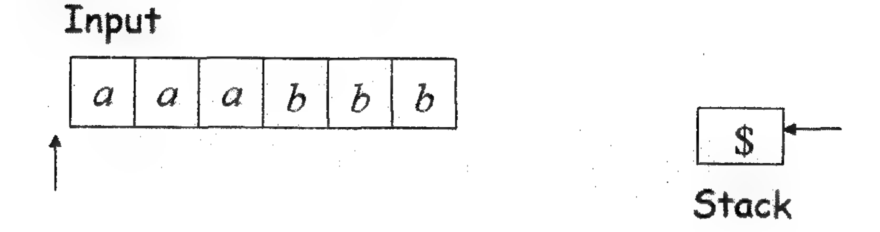
والشكل التالي يوضح مدلولات عملية الانتقال من حالة لاخرى:

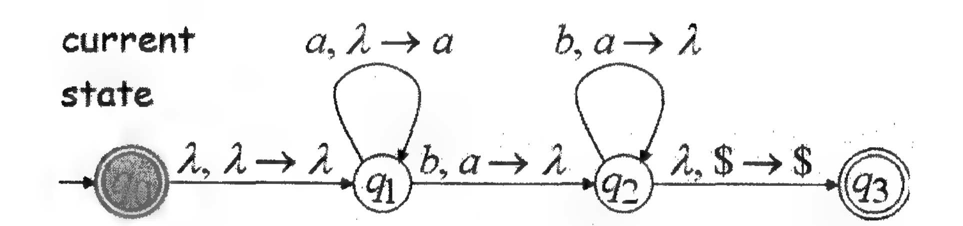


وفيما يلي خطوات تنفيذ هذه الالة وبشكل مفصل أملا في ايضاح مفهوم هذا النوع من الالات:

.1

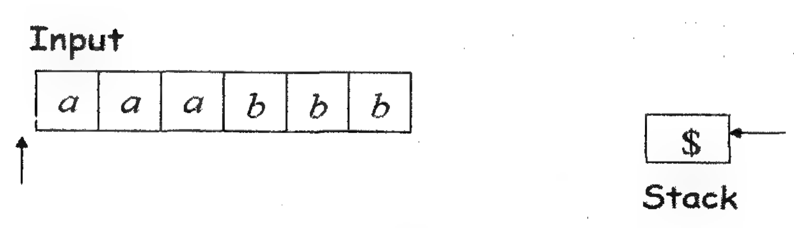
Execution Example: Time 0

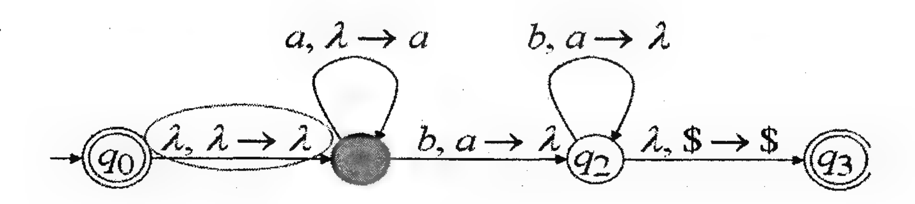




.2







.3

Time 2

Input

a	a	a	b	b	b

↑

Stack
a ←
$

Stack

$a, \lambda \rightarrow a$ $b, a \rightarrow \lambda$

$\rightarrow q_0$ $\lambda, \lambda \rightarrow \lambda$ q_1 $b, a \rightarrow \lambda$ q_2 $\lambda, \$ \rightarrow \$$ q_3

.4

Time 3

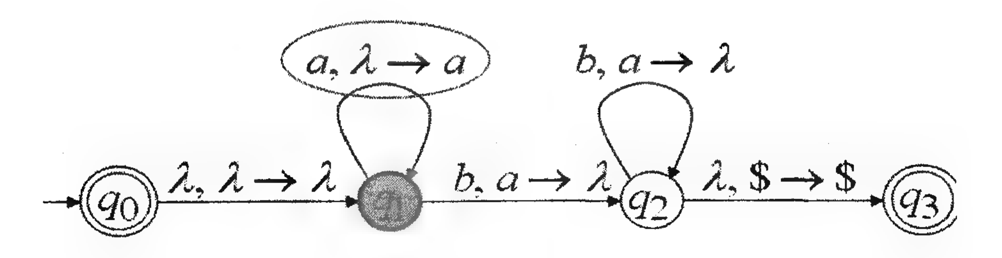
Input

a	a	a	b	b	b

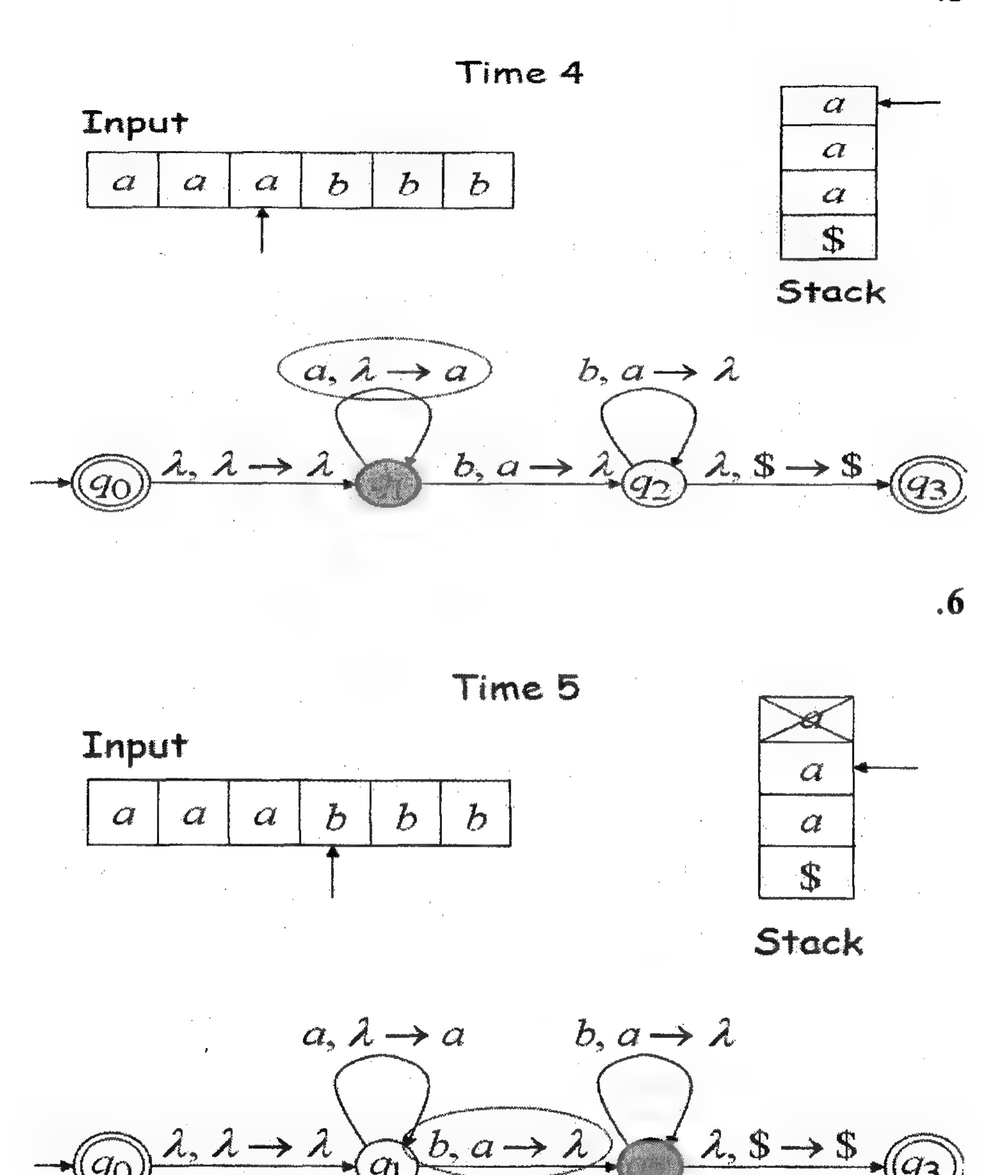
↑

Stack
a ←
a
$

Stack



.5



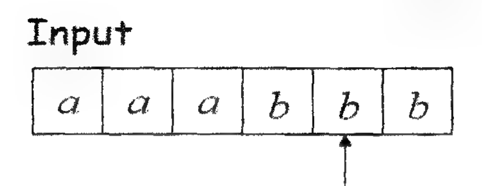
Time 4
Input
a a a b b b
a
a
a
$
Stack
a, λ → a
b, a → λ
λ, λ → λ
b, a → λ
λ, $ → $
q0
q1
q2
q3

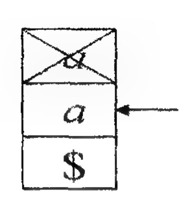
.6

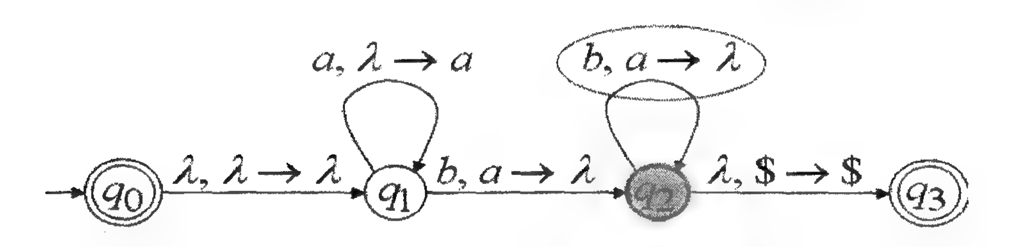
Time 5
Input
a a a b b b
a
a
a
$
Stack
a, λ → a
b, a → λ
λ, λ → λ
b, a → λ
λ, $ → $
q0
q1
q3

.7

Time 6



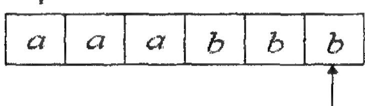


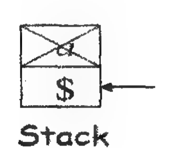


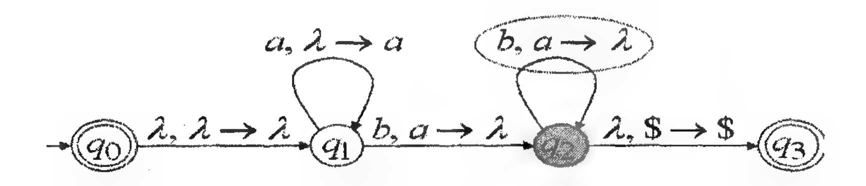
.8

Time 7



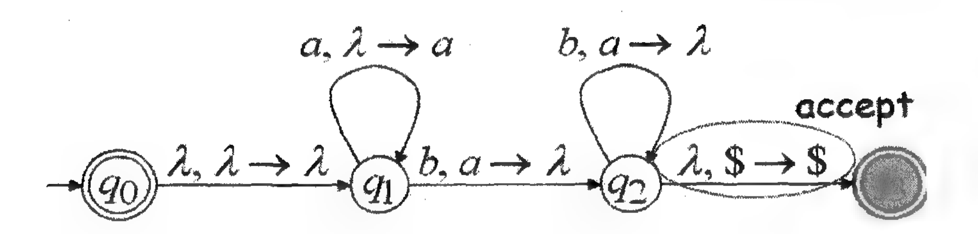






.9

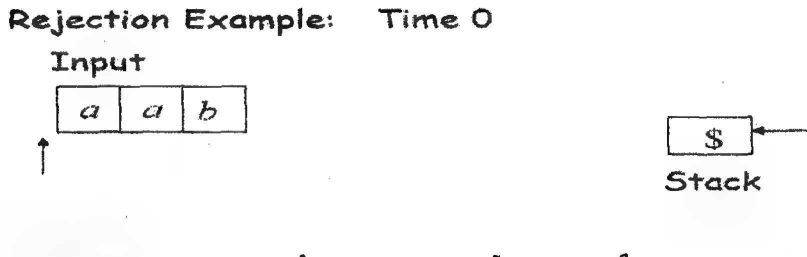
Time 8

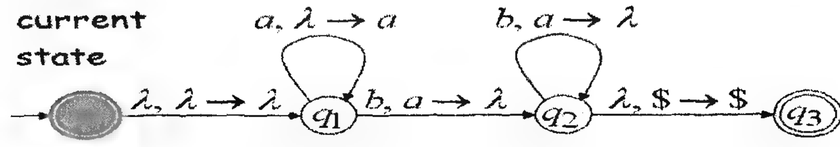


مثال:

عملية رفض الرموز

هل اللغة التالية مقبولة من قبل الالة؟

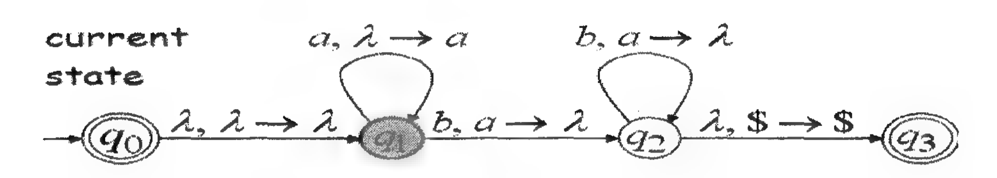




لنتتبع هذه الالة خطوة خطوة:

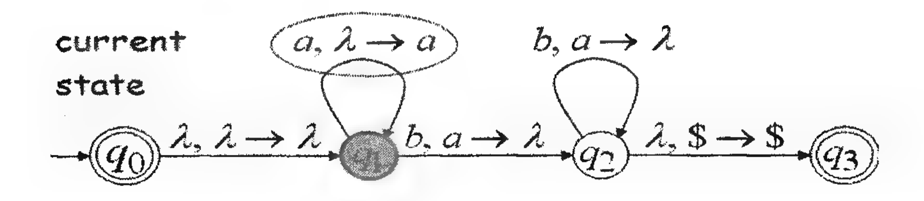
.1



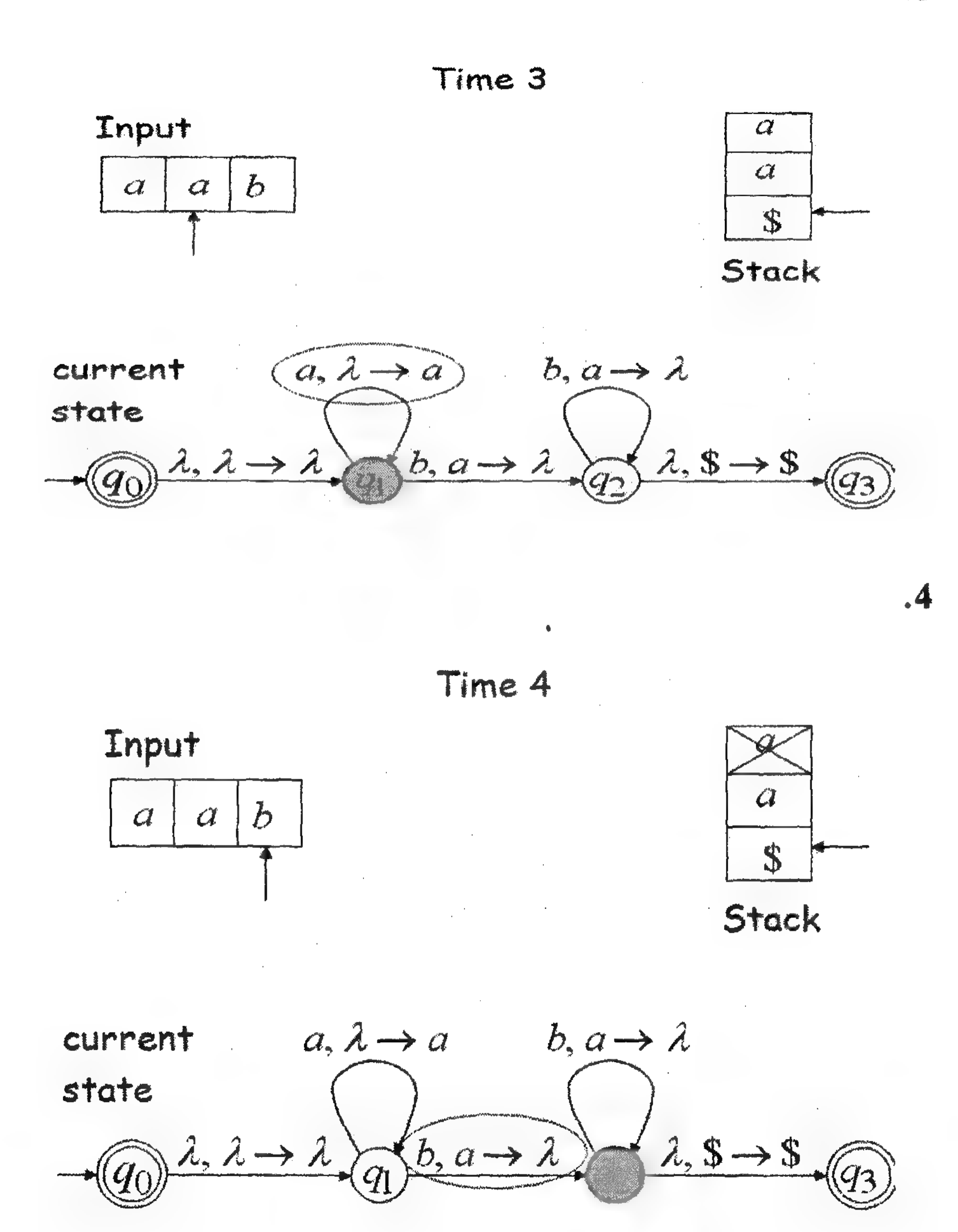


.2

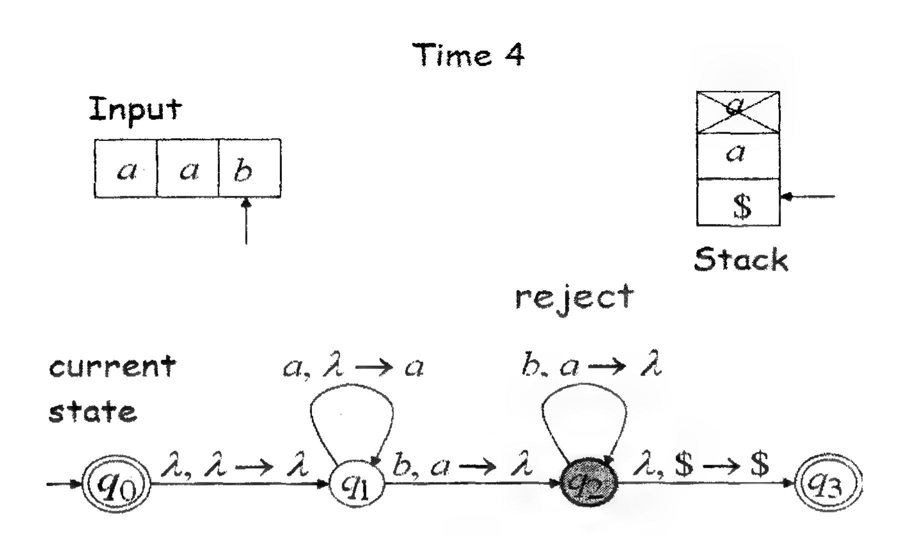




.3


Time 3
Input
a a b
a
a
$
Stack
current
state
a, λ → a
b, a → λ
λ, λ → λ
b, a → λ
λ, $ → $
q0
q1
q2
q3
.4
Time 4
Input
a a b
a
a
$
Stack
current
state
a, λ → a
b, a → λ
λ, λ → λ
b, a → λ
λ, $ → $
q0
q1
q3

.5

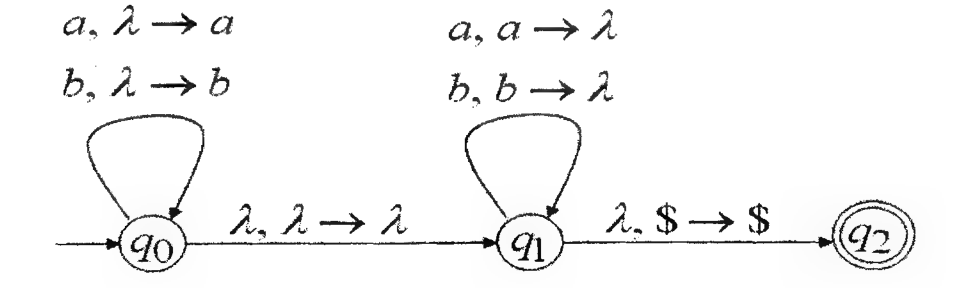


مثال:

لناخذ الالة التالية:

$$L(M) = \{vv^R : v \in \{a,b\}^*\}$$

PDA M



والشكل التالي يوضح مدلولات عملية الانتقال من حالة لأخرى:

Basic Idea: $L(M) = \{vv^R : v \in \{a, b\}^*\}$

1. Push v on stack

2. Guess middle of input

3. Match v^R on input with v on stack

4. Match found

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

$a, a \rightarrow \lambda$

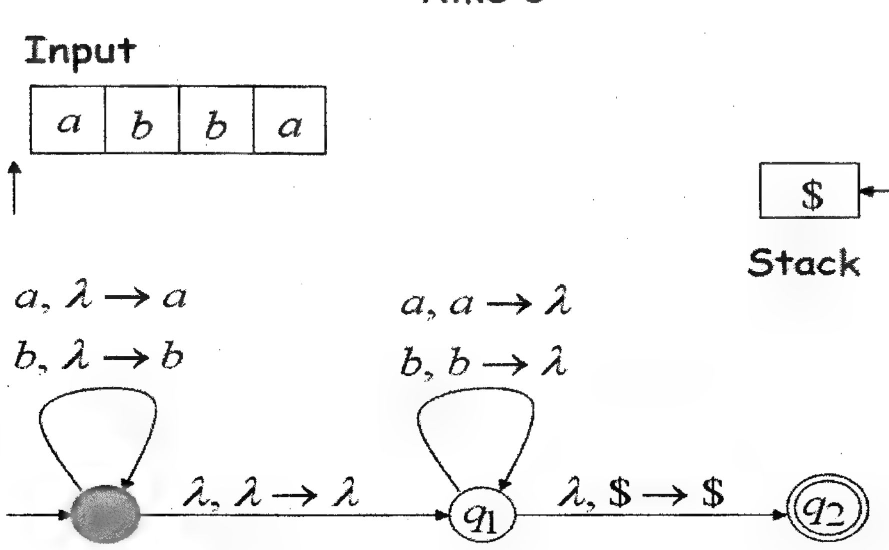
$b, b \rightarrow \lambda$

q_0 — $\lambda, \lambda \rightarrow \lambda$ → q_1 — $\lambda, \$ \rightarrow \$$ → q_2

لنتتبع تنفيذ هذه الالة خطوة خطوة

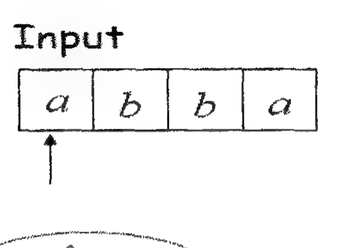
.1

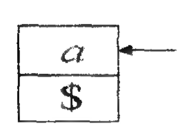


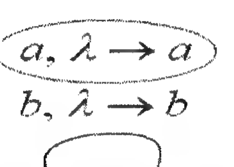


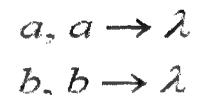
.2

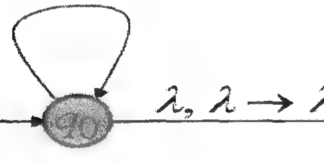

Time 1

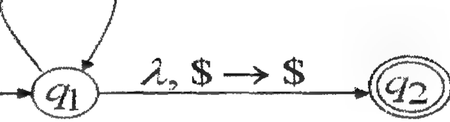

Input
a
b
b
a


a
$
Stack


a, λ → a
b, λ → b

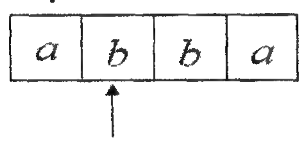

a, a → λ
b, b → λ

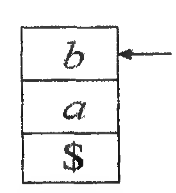

λ, λ → λ

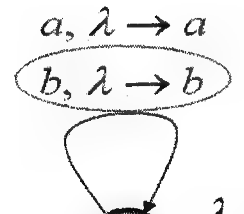

λ, $ → $

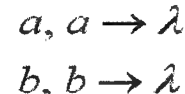
.3

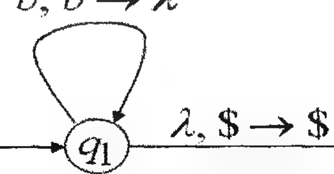

Input


a
b
b
a


b
a
$
Stack


a, λ → a
b, λ → b

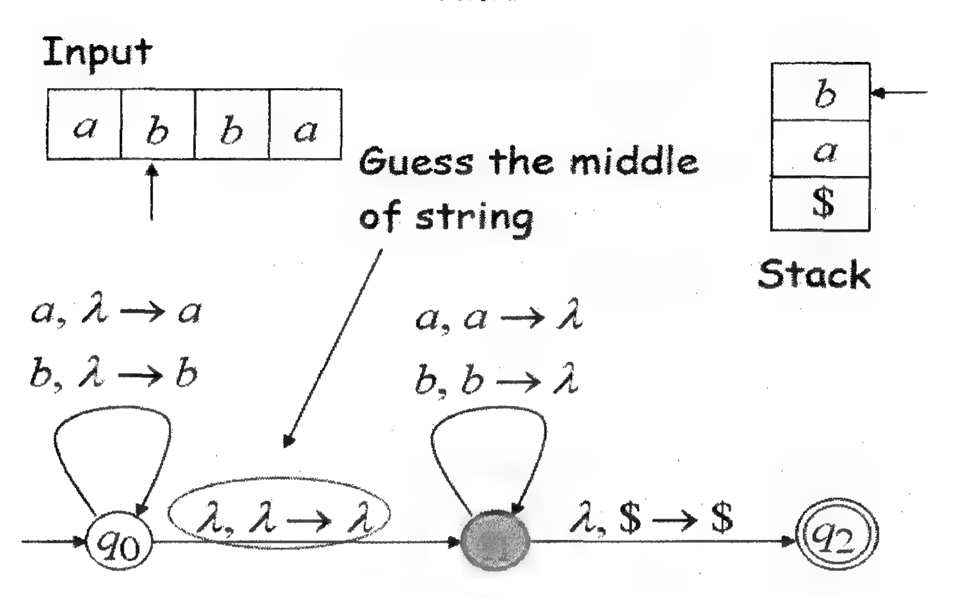

a, a → λ
b, b → λ


λ, λ → λ
λ, $ → $



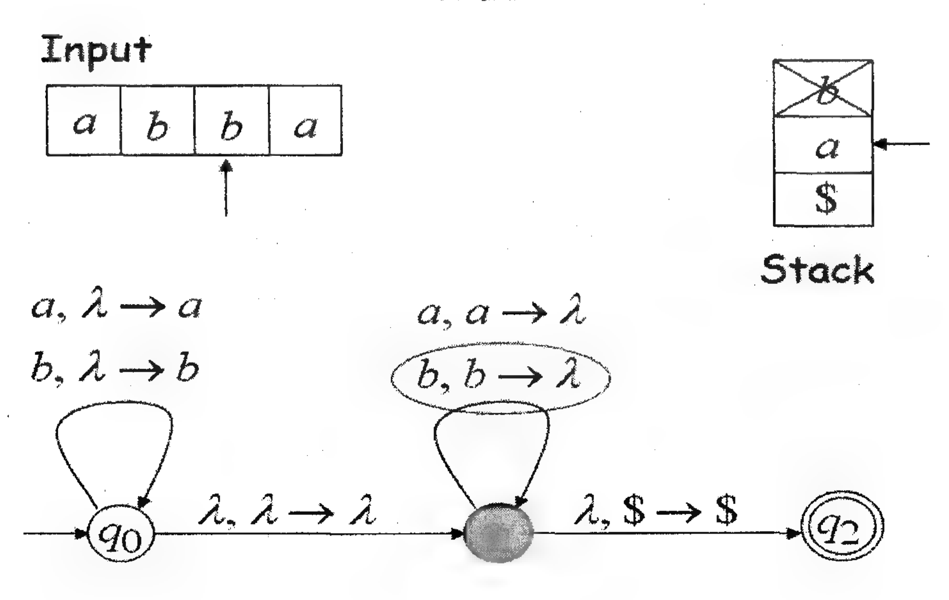
.4


Time 3


Input
a
b
b
a
Guess the middle of string
b
a
$
Stack
a, λ → a
b, λ → b
a, a → λ
b, b → λ
λ, λ → λ
λ, $ → $
q0
q2

.5

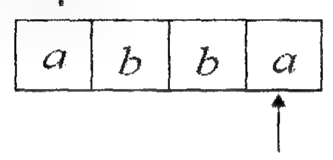
Time 4

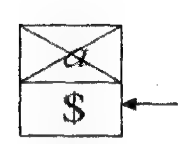

Input
a
b
b
a
b
a
$
Stack
a, λ → a
b, λ → b
a, a → λ
b, b → λ
λ, λ → λ
λ, $ → $
q0
q2

.6

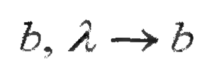
Time 5

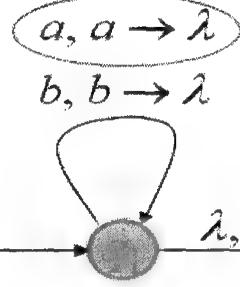
Input





$a, \lambda \rightarrow a$





$\lambda, \lambda \rightarrow \lambda$

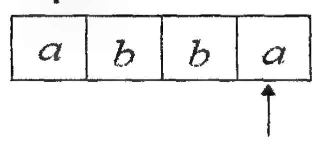


q_0

.7

Time 6

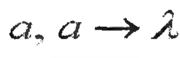
Input



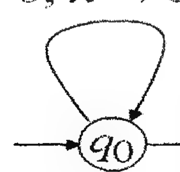
$ Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$



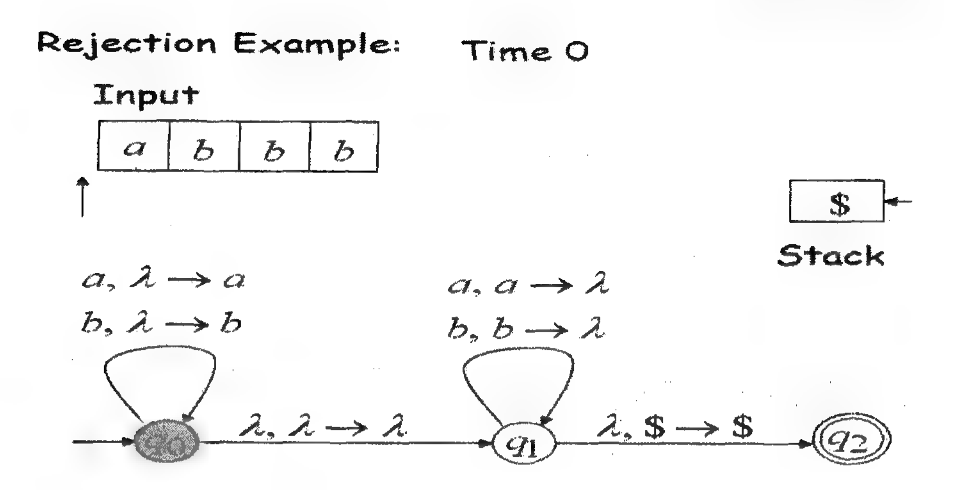
$b, b \rightarrow \lambda$



$\lambda, \lambda \rightarrow \lambda$ q_1 $\lambda, \$ \rightarrow \$$ accept q_2

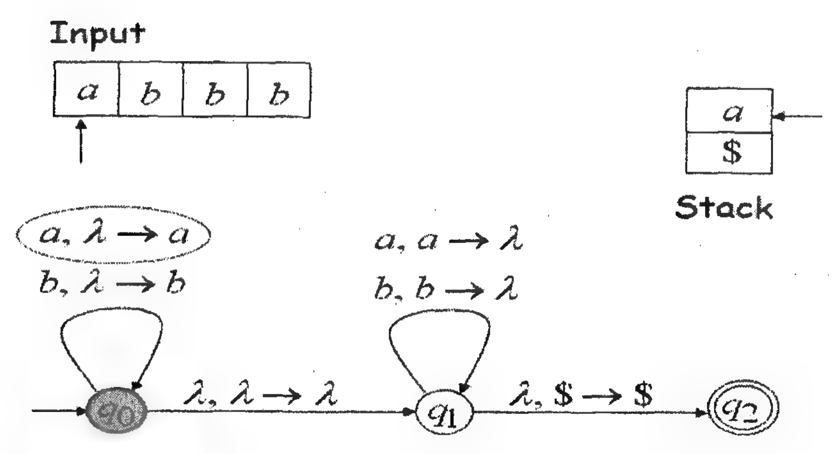
مثال:

عملية رفض الرموز



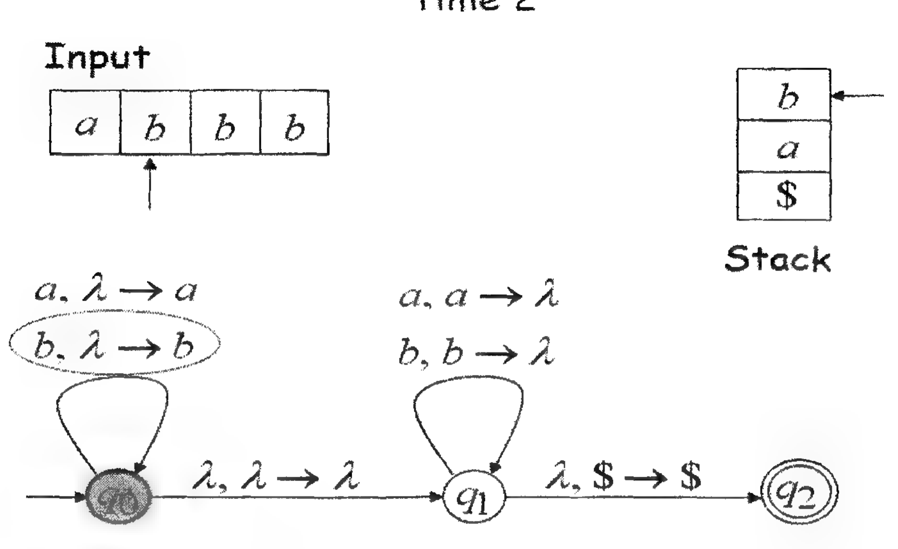
.1



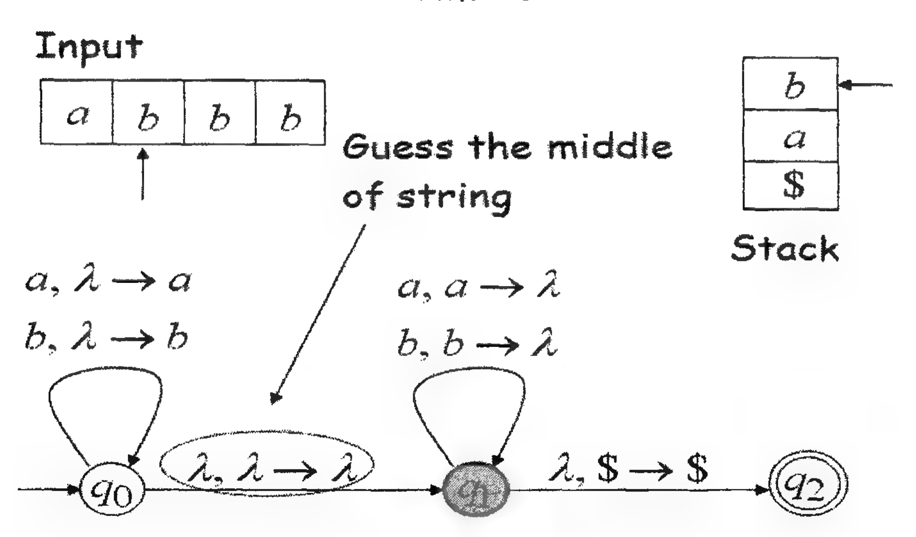


.2


Time 2

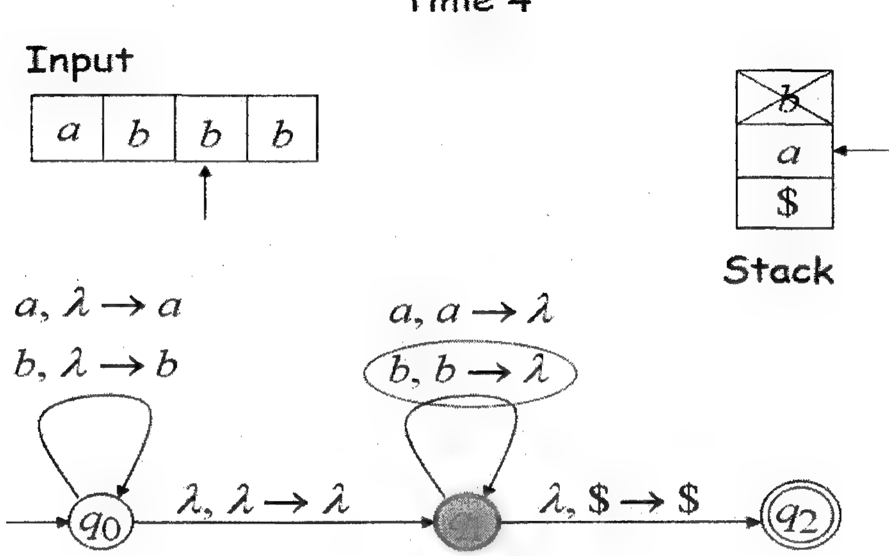

Input
a b b b
b
a
$
Stack
a, λ → a
b, λ → b
a, a → λ
b, b → λ
λ, λ → λ
λ, $ → $
q0
q1
q2

.3

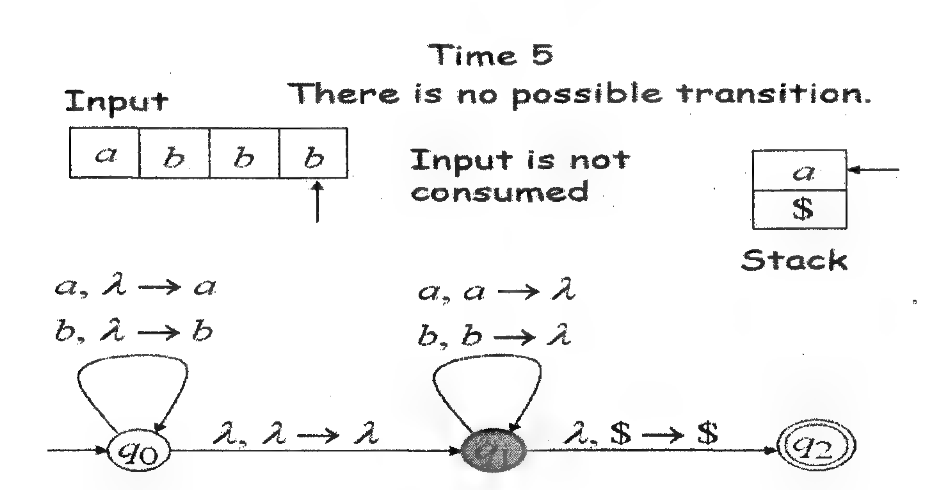

Time 3
Input
a b b b
Guess the middle of string
b
a
$
Stack
a, λ → a
b, λ → b
a, a → λ
b, b → λ
λ, λ → λ
λ, $ → $
q0
q1
q2

.4




Input
a b b b
Stack
b
a
$
a, λ → a
b, λ → b
a, a → λ
b, b → λ
λ, λ → λ
λ, $ → $
q0
q1
q2

.5


Time 5
There is no possible transition.
Input
a b b b
Input is not consumed
a
$
Stack
a, λ → a
b, λ → b
a, a → λ
b, b → λ
λ, λ → λ
λ, $ → $
q0
q1
q2

مثال آخر:

Input

Time 0

a	b	b	b

$\$$

Stack

$a, \lambda \rightarrow a$

$b, \lambda \rightarrow b$

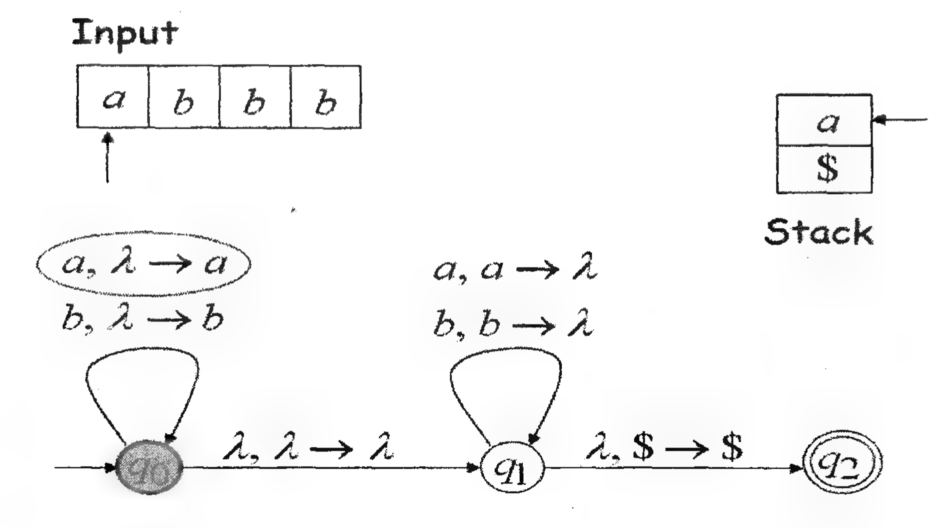
$a, a \rightarrow \lambda$

$b, b \rightarrow \lambda$

q_0 $\lambda, \lambda \rightarrow \lambda$ q_1 $\lambda, \$ \rightarrow \$$ q_2

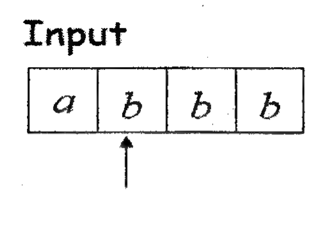
.1

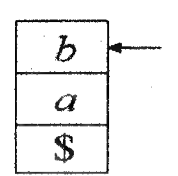
Time 1



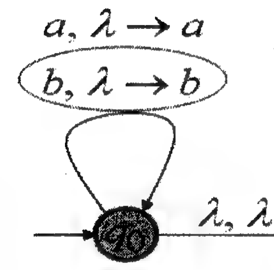
.2

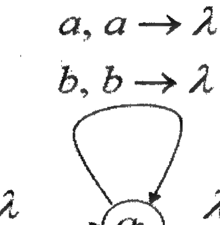

Time 2

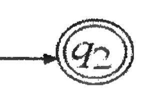

Input
a
b
b
b


b
a
$

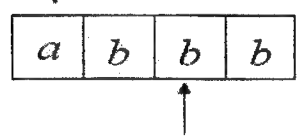

Stack

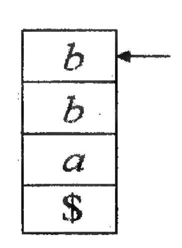

a, λ → a
b, λ → b
q0
λ, λ → λ

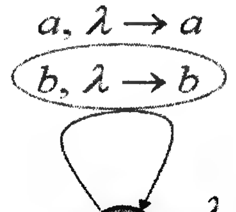

a, a → λ
b, b → λ
q1
λ, $ → $

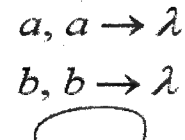

q2

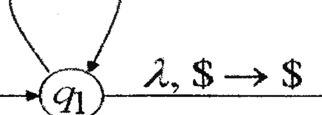
.3


Time 3
Input
a
b
b
b


b
b
a
$
Stack

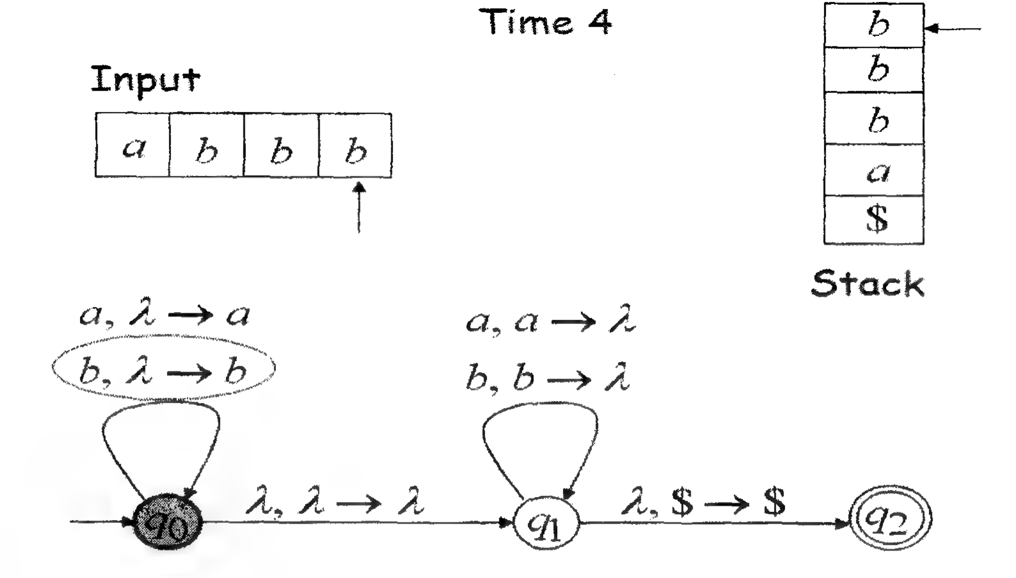

a, λ → a
b, λ → b
q0
λ, λ → λ


a, a → λ
b, b → λ

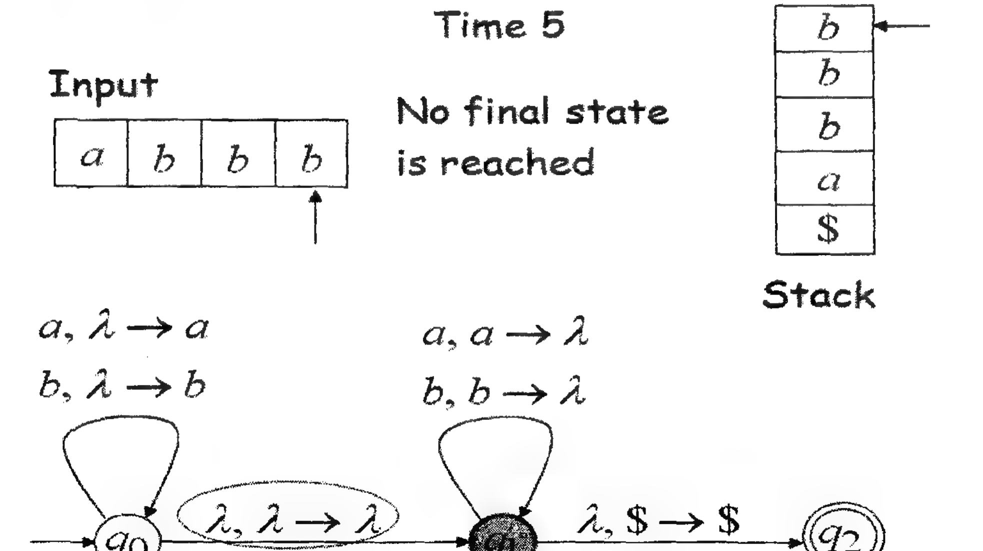

q1
λ, $ → $
q2



.4

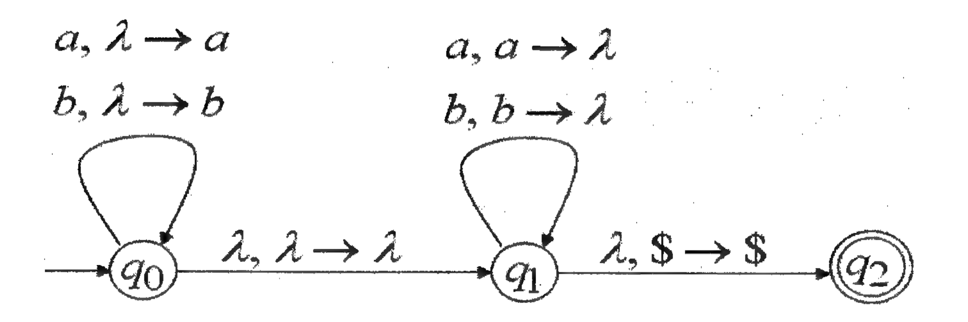

Time 4
Input
a b b b
b
b
b
a
$
Stack
a, λ → a
b, λ → b
a, a → λ
b, b → λ
λ, λ → λ
λ, $ → $
q0
q1
q2

.5


Time 5
Input
a b b b
No final state
is reached
b
b
b
a
$
Stack
a, λ → a
b, λ → b
a, a → λ
b, b → λ
λ, λ → λ
λ, $ → $
q0
q1
q2

There is no computation
that accepts string $abbb$

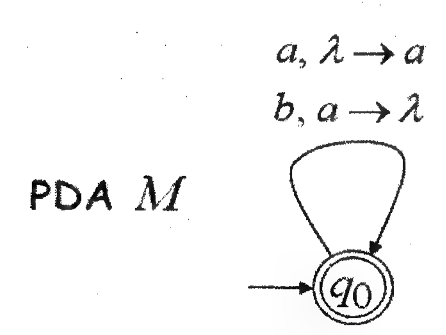
$$abbb \notin L(M)$$



مثال:

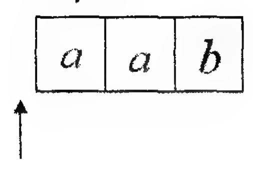
$$L(M) = \{w \in \{a,b\}^* :$$
$$\text{in every prefix } v, \quad n_a(v) \geq n_b(v)\}$$

PDA M

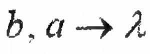


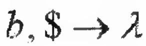
Execution Example: Time 0

Input











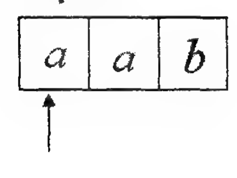
$

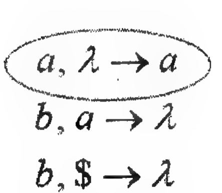
Stack

.1

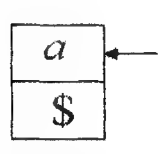
Time 1

Input





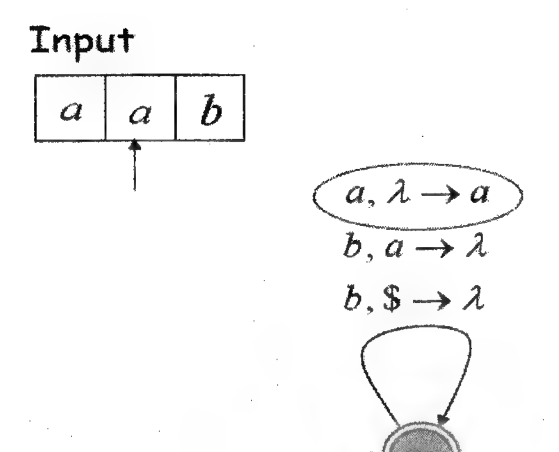


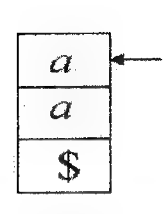


Stack

.2

Time 2



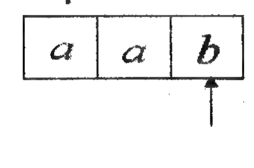


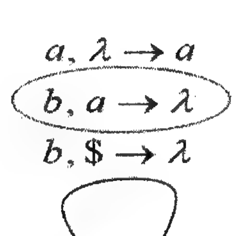
Stack

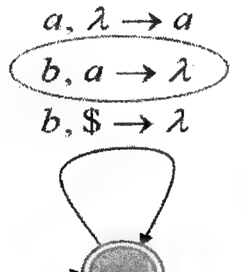
.3

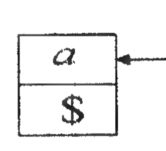
Time 3

Input







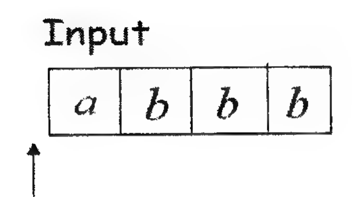


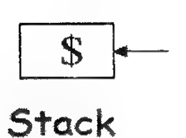
Stack

accept

عملية الرفض:

Rejection example: Time 0



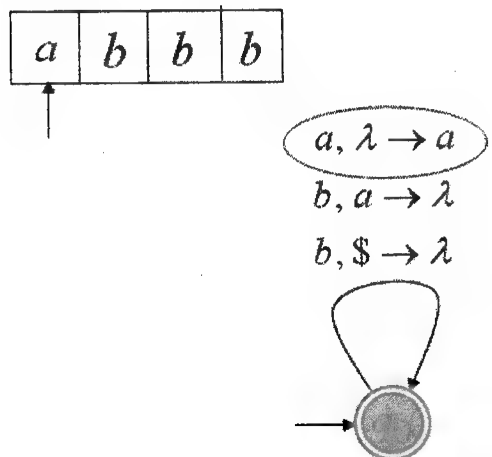


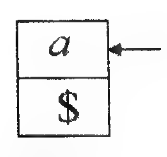


.1

Time 1

Input

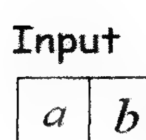


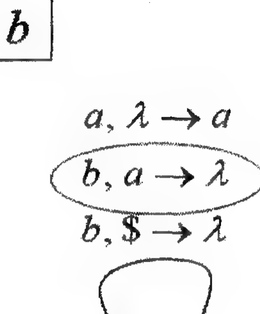


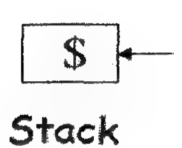
Stack

.2

Time 2



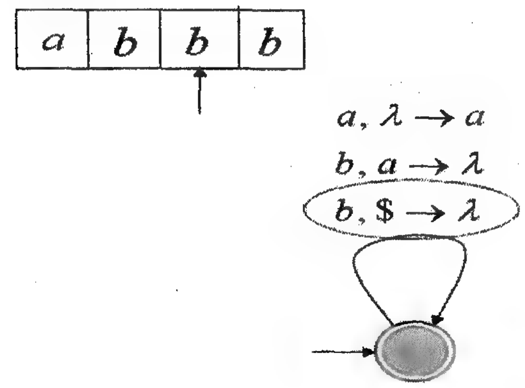






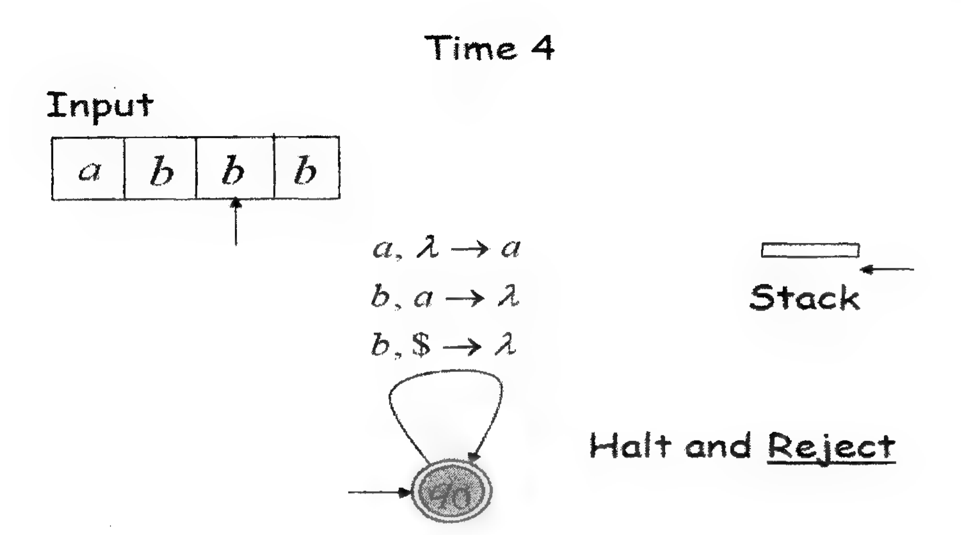
.3

Time 3

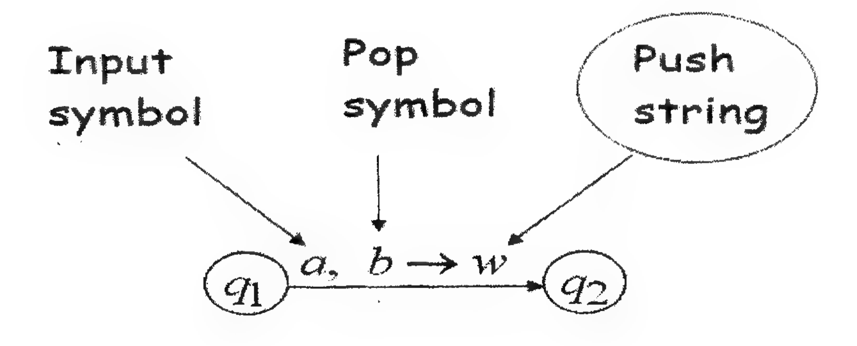




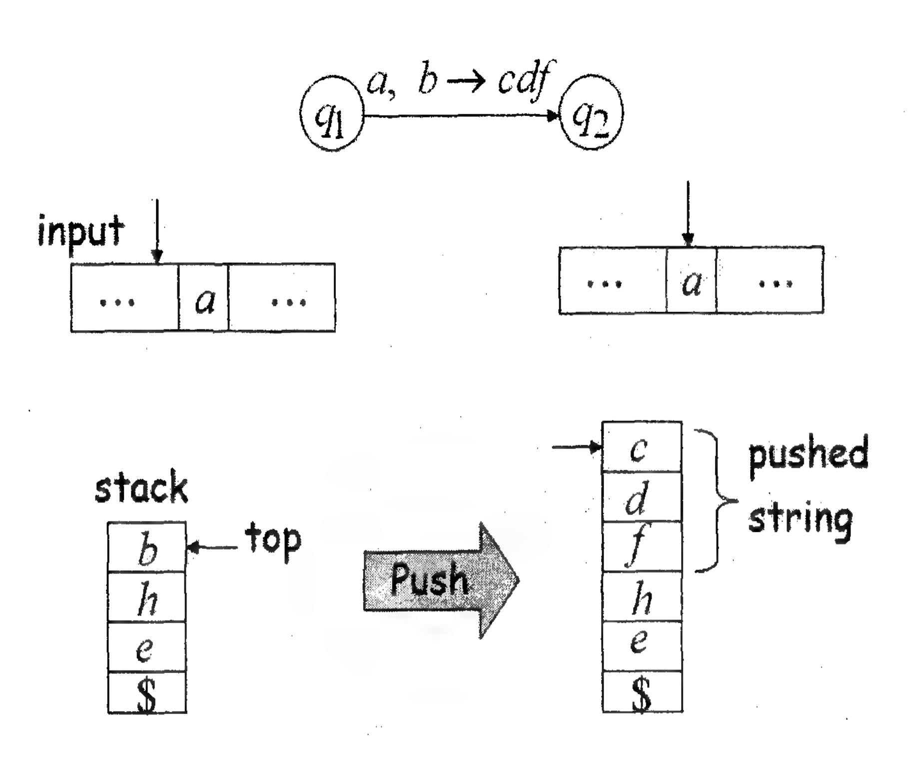
.4



اشرنا سابقا الى إن عملية الإنتقال من حالة لحالة تصاحبها عملية اضافة رمز الى الحزمة هذا ويمكن تنفيذ عملية الاضافة لاكثر من رمز عند النتقال من حالة لأخرى بحيث تخزن هذه الرموز في الحزمة تباعا ويبين الشكل التالي تمثيل عملية الانتقال من حالة لاخرى بتسجيل أو اضافة مجموعة من الرموز الى الحزمة:



والمثال التالي يوضح هذه العملية:



مثال:

$$a, \$ \rightarrow 0\$ \qquad b, \$ \rightarrow 1\$$$

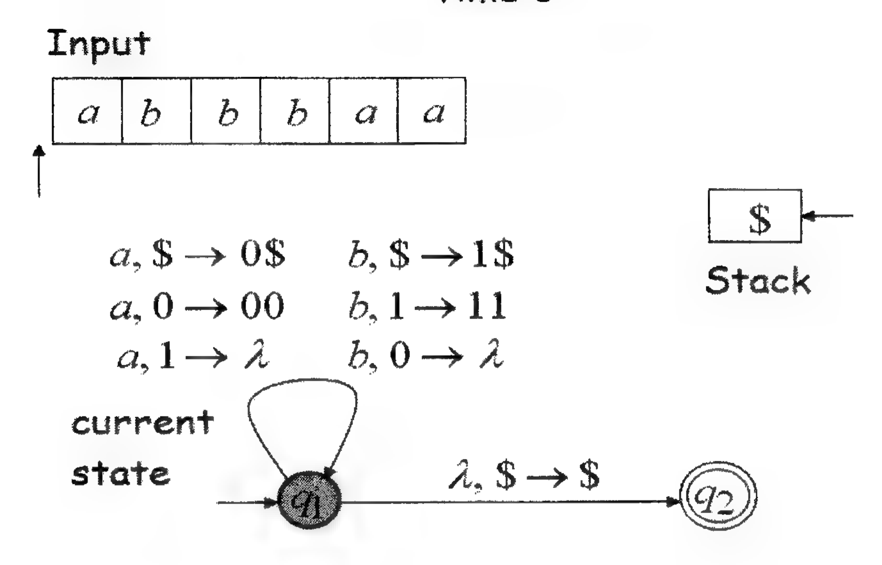
$$a, 0 \rightarrow 00 \qquad b, 1 \rightarrow 11$$

$$a, 1 \rightarrow \lambda \qquad b, 0 \rightarrow \lambda$$

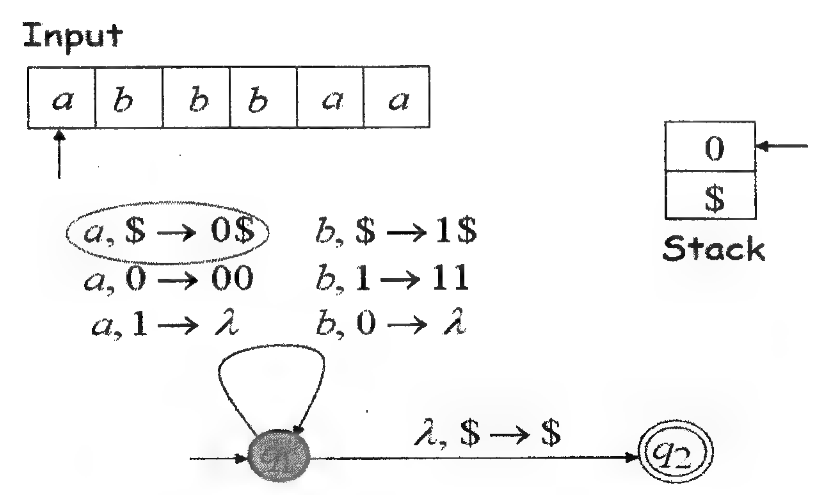
$$\lambda, \$ \rightarrow \$$$

q_1 q_2

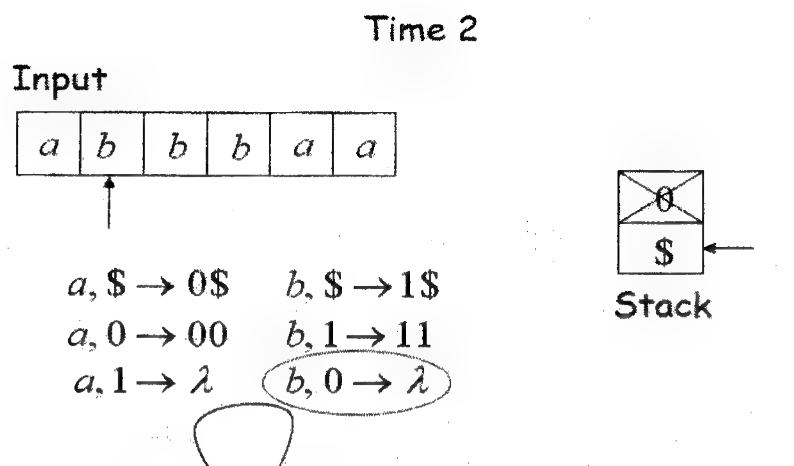
لنتتبع هذه الالة باخذ مجموعة من الرموز



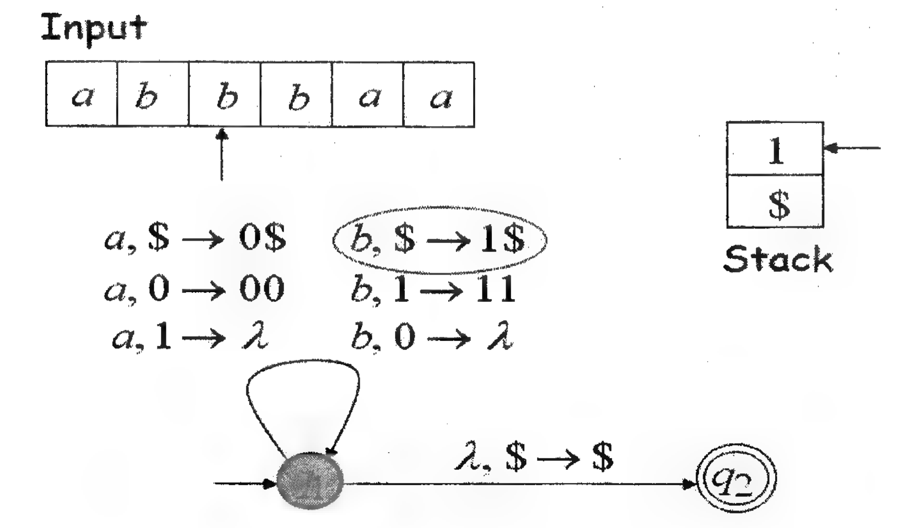
.1



.2


Time 2
Input
a b b b a a
0
$
Stack
a, $ → 0$
a, 0 → 00
a, 1 → λ
b, $ → 1$
b, 1 → 11
b, 0 → λ
λ, $ → $
q1
q2

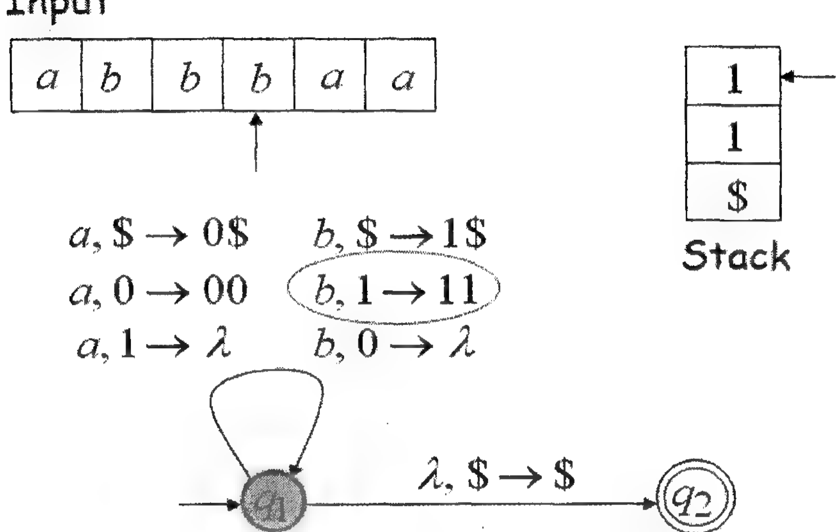
.3


Time 3
Input
a b b b a a
1
$
Stack
a, $ → 0$
a, 0 → 00
a, 1 → λ
b, $ → 1$
b, 1 → 11
b, 0 → λ
λ, $ → $
q1
q2

.4

Time 4

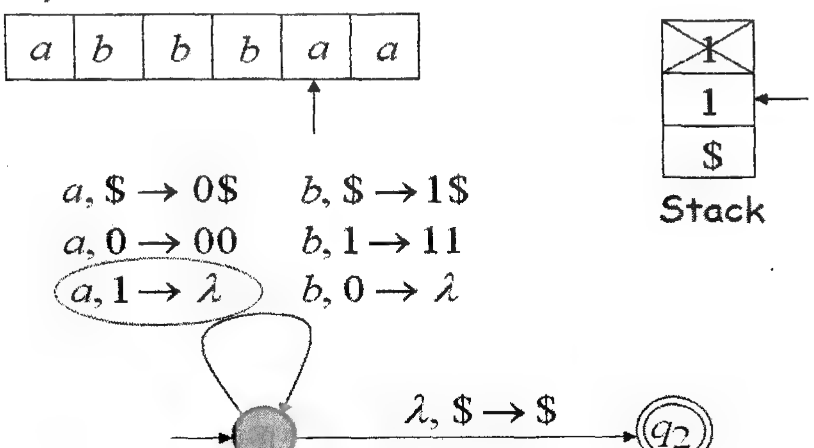




.5

Time 5

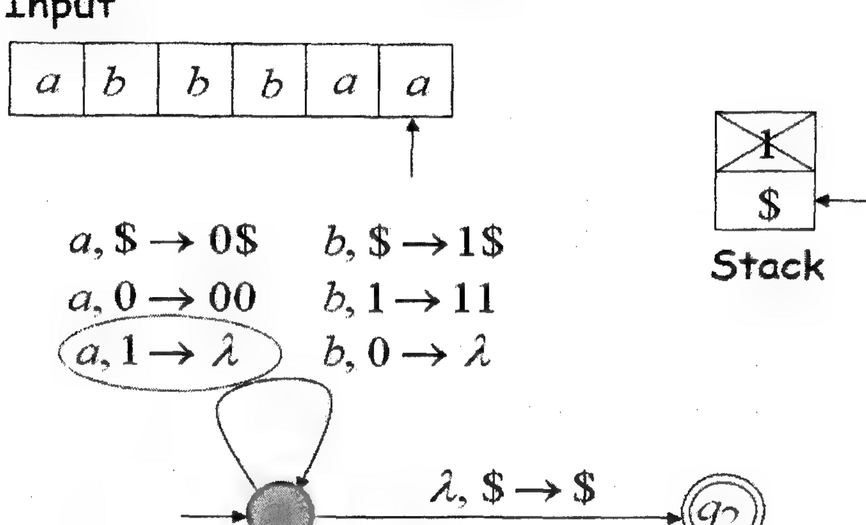
Input



.6



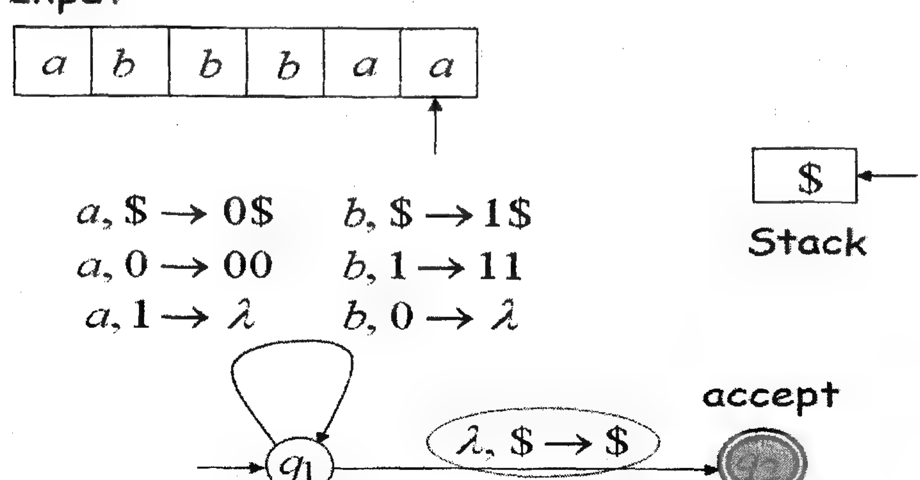




.7

Time 7

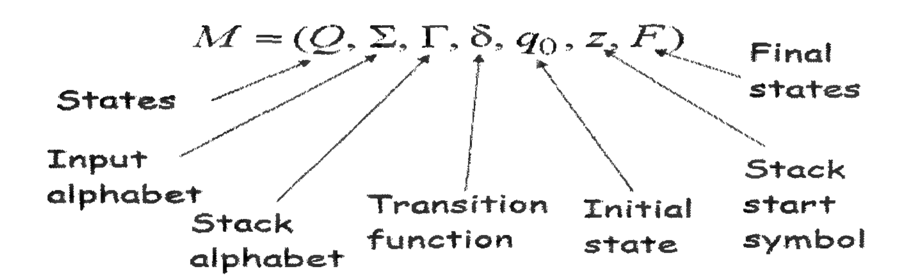
Input



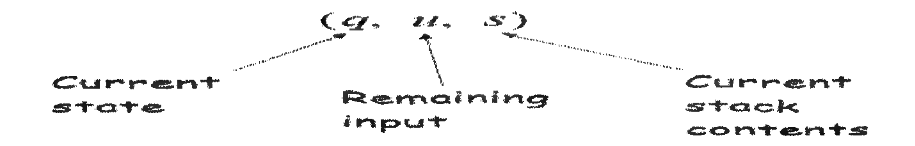
2.4 التعريف الشكلي لالة الحالة المنتهية المستخدمة للحزمة:

يضم النموذج الرياضي لآلة الحالة النتهية المستخدمة للحزمة وكما هو مبين في الشكل التالي مجموعة من المكونات هي:

- مجموعة الحالات بما فيها الحالة الابتدائية والحالات النهائية والمرحلية.
- مجموعة رموز المدخلات والمراد التعرف عليها أو رفضها.
- مجموعة رموز الحزمة.
- دالة الانتقال.
- الحالة الابتدائية.
- رمز البداية للحزمة.
- مجموعة الحالات النهائية.



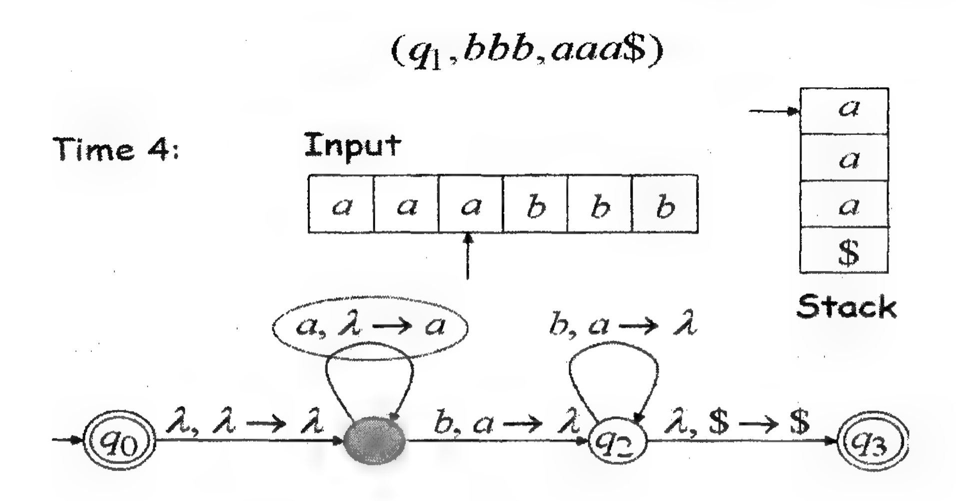
اما هيئة الالة أو عملية الوصف الانية فتضم:



- الحالة الحالية.
- الرموز المتبقية للقراءة.
- محتوى الحزمة الحالي من الرموز.

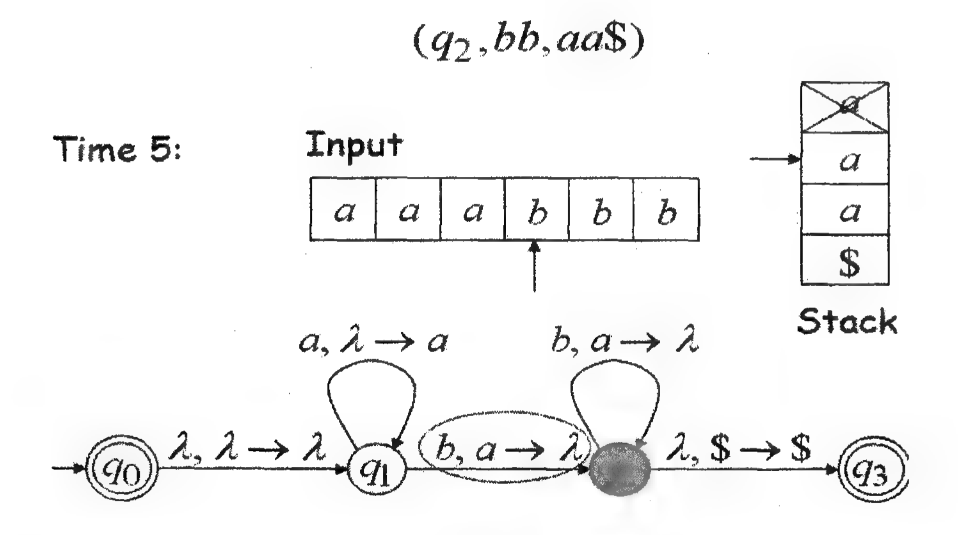
والشكل التالي يبين هيئة الالة في لحظة زمنية معينة:

$(q_1, bbb, aaa\$)$



وتتغير الهيئة من عملية انتقال الى اخرى لتغير الحالة وحالة الشريط وحالة الحزمة ويبين الشكل التالي هيئة الالة اللاحقة للهيئة المبينة اعلاه:

$(q_2, bb, aa\$)$



وتمثل عملية الانتقال من هيئة في الحظة الرابعة الى هيئة جديدة في الحظة الخامسة كما يلي:

$$(q_1, bbb, aaa\$) \succ (q_2, bb, aa\$)$$

Time 4 Time 5

ويمكن استخدام الهيئات لحساب الالة كما يلي:

$$(q_0, aaabbb, \$) \succ (q_1, aaabbb, \$) \succ$$
$$(q_1, aabbb, a\$) \succ (q_1, abbb, aa\$) \succ (q_1, bbb, aaa\$) \succ$$
$$(q_2, bb, aa\$) \succ (q_2, b, a\$) \succ (q_2, \lambda, \$) \succ (q_3, \lambda, \$)$$

ولتسهيل عملية التمثيل يمكن اختصار عملية الحساب هذه كما يلي:

$$(q_0, aaabbb, \$) \overset{*}{\succ} (q_3, \lambda, \$)$$

وبهذا بمكن تعريف الالة المنتهية باستخدام مفهوم الهيئة على انها الة تبدء بهيئة ابتدائية وتنتهي بهيئة نهائية وكما هو مبين أدناه حيث تشكل عملية الانتقال هذه لغة الالة:

$$L(M) = \{w : (q_0, w, s) \overset{*}{\succ} (q_f, \lambda, s')\}$$

Initial state Final state

مثال:

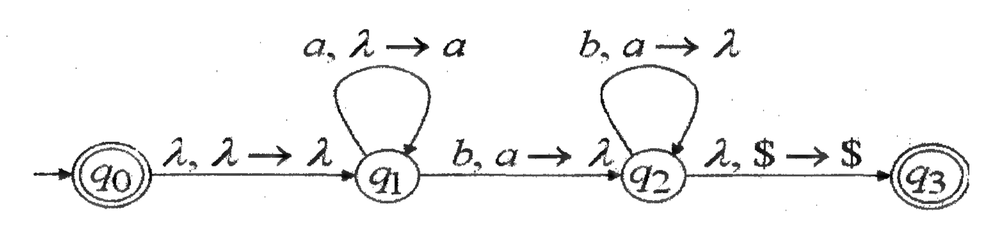
Example:

$$(q_0, aaabbb, \$) \overset{*}{\succ} (q_3, \lambda, \$)$$

$$\Downarrow$$

$$aaabbb \in L(M)$$

PDA M:



$$(q_0, a^n b^n, \$) \overset{*}{\succ} (q_3, \lambda, \$)$$

$$\Downarrow$$

$$a^n b^n \in L(M)$$

PDA M:

$a, \lambda \rightarrow a$ $b, a \rightarrow \lambda$

q_0 $\lambda, \lambda \rightarrow \lambda$ q_1 $b, a \rightarrow \lambda$ q_2 $\lambda, \$ \rightarrow \$$ q_3

3.4تمارين:

1. صمم الالة المنتهية لموازنة الاقواس الدئرية المفتوحة والأقواس الدئرية المغلقة.

الحل:

$M = (\{q_1\}, \{“(“, “)”\}, \{L, \#\}, \delta, q_1, \#, \emptyset)$

δ:

(1) $\delta (q_1, (, \#) = \{(q_1, L\#)\}$

(2) $\delta (q_1,), \#) = \emptyset$

(3) $\delta (q_1, (, L) = \{(q_1, LL)\}$

(4) $\delta (q_1,), L) = \{(q_1, \varepsilon)\}$

(5) $\delta (q_1, \varepsilon, \#) = \{(q_1, \varepsilon)\}$

(6) $\delta (q_1, \varepsilon, L) = \emptyset$

- **Transition Diagram:**

(, # | L#

ε, # | ε q_0 (, L | LL

), L | ε

- **Example Computation:**

Current Input	Stack	Transition	
(())	#		
())	L#	(1)	- Could have applied rule (5), but it would have done no good
))	LL#	(3)	
)	L#	(4)	
ε	#	(4)	
ε	-	(5)	

.2

- **Example Computation:**

(1) $\delta(q_1, 0, R) = \{(q_1, BR)\}$
(2) $\delta(q_1, 0, B) = \{(q_1, BB)\}$
(3) $\delta(q_1, 0, G) = \{(q_1, BG)\}$
(4) $\delta(q_1, c, R) = \{(q_2, R)\}$
(5) $\delta(q_1, c, B) = \{(q_2, B)\}$
(6) $\delta(q_1, c, G) = \{(q_2, G)\}$
(7) $\delta(q_2, 0, B) = \{(q_2, \varepsilon)\}$
(8) $\delta(q_2, \varepsilon, R) = \{(q_2, \varepsilon)\}$
(9) $\delta(q_1, 1, R) = \{(q_1, GR)\}$
(10) $\delta(q_1, 1, B) = \{(q_1, GB)\}$
(11) $\delta(q_1, 1, G) = \{(q_1, GG)\}$
(12) $\delta(q_2, 1, G) = \{(q_2, \varepsilon)\}$

State	Input	Stack	Rule Applied	Rules Applicable
q_1	01c10	R		(1)
q_1	1c10	BR	(1)	(10)
q_1	c10	GBR	(10)	(6)
q_2	10	GBR	(6)	(12)
q_2	0	BR	(12)	(7)
q_2	ε	R	(7)	(8)
q_2	ε	ε	(8)	-

- **Example Computation:**

(1) $\delta(q_1, 0, R) = \{(q_1, BR)\}$
(2) $\delta(q_1, 0, B) = \{(q_1, BB)\}$
(3) $\delta(q_1, 0, G) = \{(q_1, BG)\}$
(4) $\delta(q_1, c, R) = \{(q_2, R)\}$
(5) $\delta(q_1, c, B) = \{(q_2, B)\}$
(6) $\delta(q_1, c, G) = \{(q_2, G)\}$
(7) $\delta(q_2, 0, B) = \{(q_2, \varepsilon)\}$
(8) $\delta(q_2, \varepsilon, R) = \{(q_2, \varepsilon)\}$
(9) $\delta(q_1, 1, R) = \{(q_1, GR)\}$
(10) $\delta(q_1, 1, B) = \{(q_1, GB)\}$
(11) $\delta(q_1, 1, G) = \{(q_1, GG)\}$
(12) $\delta(q_2, 1, G) = \{(q_2, \varepsilon)\}$

State	Input	Stack	Rule Applied
q_1	1c1	R	
q_1	c1	GR	(9)
q_2	1	GR	(6)
q_2	ε	R	(12)
q_2	ε	ε	(8)

مثال:

- **Example PDA** : For the language $\{x \mid x = ww^r \text{ and } w \text{ in } \{0,1\}^*\}$

$M = (\{q_1, q_2\}, \{0, 1\}, \{R, B, G\}, \delta, q_1, R, \emptyset)$

δ:

(1) $\delta(q_1, 0, R) = \{(q_1, BR)\}$
(2) $\delta(q_1, 1, R) = \{(q_1, GR)\}$
(3) $\delta(q_1, 0, B) = \{(q_1, BB), (q_2, \varepsilon)\}$
(4) $\delta(q_1, 0, G) = \{(q_1, BG)\}$
(5) $\delta(q_1, 1, B) = \{(q_1, GB)\}$
(6) $\delta(q_1, 1, G) = \{(q_1, GG), (q_2, \varepsilon)\}$
(7) $\delta(q_2, 0, B) = \{(q_2, \varepsilon)\}$
(8) $\delta(q_2, 1, G) = \{(q_2, \varepsilon)\}$
(9) $\delta(q_1, \varepsilon, R) = \{(q_2, \varepsilon)\}$
(10) $\delta(q_2, \varepsilon, R) = \{(q_2, \varepsilon)\}$

- **Notes:**
  - Rules #3 and #6 are non-deterministic.
  - Rules #9 and #10 are used to pop the final stack symbol off at the end of a computation.

- **Example Computation:**

(1) $\delta(q_1, 0, R) = \{(q_1, BR)\}$
(2) $\delta(q_1, 1, R) = \{(q_1, GR)\}$
(3) $\delta(q_1, 0, B) = \{(q_1, BB), (q_2, \varepsilon)\}$
(4) $\delta(q_1, 0, G) = \{(q_1, BG)\}$
(5) $\delta(q_1, 1, B) = \{(q_1, GB)\}$
(6) $\delta(q_1, 1, G) = \{(q_1, GG), (q_2, \varepsilon)\}$
(7) $\delta(q_2, 0, B) = \{(q_2, \varepsilon)\}$
(8) $\delta(q_2, 1, G) = \{(q_2, \varepsilon)\}$
(9) $\delta(q_1, \varepsilon, R) = \{(q_2, \varepsilon)\}$
(10) $\delta(q_2, \varepsilon, R) = \{(q_2, \varepsilon)\}$

State	Input	Stack	Rule Applied	Rules Applicable
q_1	000000	R		(1), (9)
q_1	00000	BR	(1)	(3), both options
q_1	0000	BBR	(3) option #1	(3), both options
q_1	000	BBBR	(3) option #1	(3), both options
q_2	00	BBR	(3)option #2	(7)
q_2	0	BR	(7)	(7)
q_2	ε	R	(7)	(10)
q_2	ε	ε	(10)	

- **Example Computation:**

(1) $\delta(q_1, 0, R) = \{(q_1, BR)\}$
(2) $\delta(q_1, 1, R) = \{(q_1, GR)\}$
(3) $\delta(q_1, 0, B) = \{(q_1, BB), (q_2, \varepsilon)\}$
(4) $\delta(q_1, 0, G) = \{(q_1, BG)\}$
(5) $\delta(q_1, 1, B) = \{(q_1, GB)\}$
(6) $\delta(q_1, 1, G) = \{(q_1, GG), (q_2, \varepsilon)\}$
(7) $\delta(q_2, 0, B) = \{(q_2, \varepsilon)\}$
(8) $\delta(q_2, 1, G) = \{(q_2, \varepsilon)\}$
(9) $\delta(q_1, \varepsilon, R) = \{(q_2, \varepsilon)\}$
(10) $\delta(q_2, \varepsilon, R) = \{(q_2, \varepsilon)\}$

State	Input	Stack	Rule Applied	
q_1	010010	R		
q_1	10010	BR	(1)	From (1) and (9)
q_1	0010	GBR	(5)	
q_1	010	BGBR	(4)	
q_2	10	GBR	(3) option #2	
q_2	0	BR	(8)	
q_2	ε	R	(7)	
q_2	ε	ε	(10)	

مثال:

- **Example** : Consider the following CFG in GNF.

(1) S -> aA
(2) S -> aB
(3) A -> aA G is in GNF
(4) A -> aB L(G) = a+b+
(5) B -> bB
(6) B -> b

Construct M as:
Q = {q}
Σ = T = {a, b}
Γ = V = {S, A, B}
z = S

(1) δ(q, a, S) = {(q, A), (q, B)} From productions #1 and 2, S->aA, S->aB
(2) δ(q, a, A) = {(q, A), (q, B)} From productions #3 and 4, A->aA, A->aB
(3) δ(q, a, B) = Ø
(4) δ(q, b, S) = Ø
(5) δ(q, b, A) = Ø
(6) δ(q, b, B) = {(q, B), (q, ε)} From productions #5 and 6, B->bB, B->b
(7) δ(q, ε, S) = Ø
(8) δ(q, ε, A) = Ø
(9) δ(q, ε, B) = Ø Recall δ: Q x (Σ U {ε}) x Γ -> finite subsets of Q x Γ*

مثال:

- **Example** : Consider the following CFG in GNF.

(1) S -> aABC
(2) A -> a　　　　G is in GNF
(3) B -> b
(4) C -> cAB
(5) C -> cC

Construct M as:

$Q = \{q\}$
$\Sigma = T = \{a, b, c\}$
$\Gamma = V = \{S, A, B, C\}$
$z = S$

(1) $\delta(q, a, S) = \{(q, ABC)\}$　S->aABC
(2) $\delta(q, a, A) = \{(q, \varepsilon)\}$　A->a
(3) $\delta(q, a, B) = \emptyset$
(4) $\delta(q, a, C) = \emptyset$
(5) $\delta(q, b, S) = \emptyset$
(6) $\delta(q, b, A) = \emptyset$
(7) $\delta(q, b, B) = \{(q, \varepsilon)\}$　B->b
(8) $\delta(q, b, C) = \emptyset$
(9) $\delta(q, c, S) = \emptyset$
(10) $\delta(q, c, A) = \emptyset$
(11) $\delta(q, c, B) = \emptyset$
(12) $\delta(q, c, C) = \{(q, AB), (q, C)\}$　C->cAB|cC
(13) $\delta(q, \varepsilon, S) = \emptyset$
(14) $\delta(q, \varepsilon, A) = \emptyset$
(15) $\delta(q, \varepsilon, B) = \emptyset$
(16) $\delta(q, \varepsilon, C) = \emptyset$

مثال:

- **Example:**

Consider L = {ε, b, ab, aab, aaab, ...} Then L' = {b, ab, aab, aaab, ...}

- **The GNF CFG for L':**

(1) S -> aS
(2) S -> b

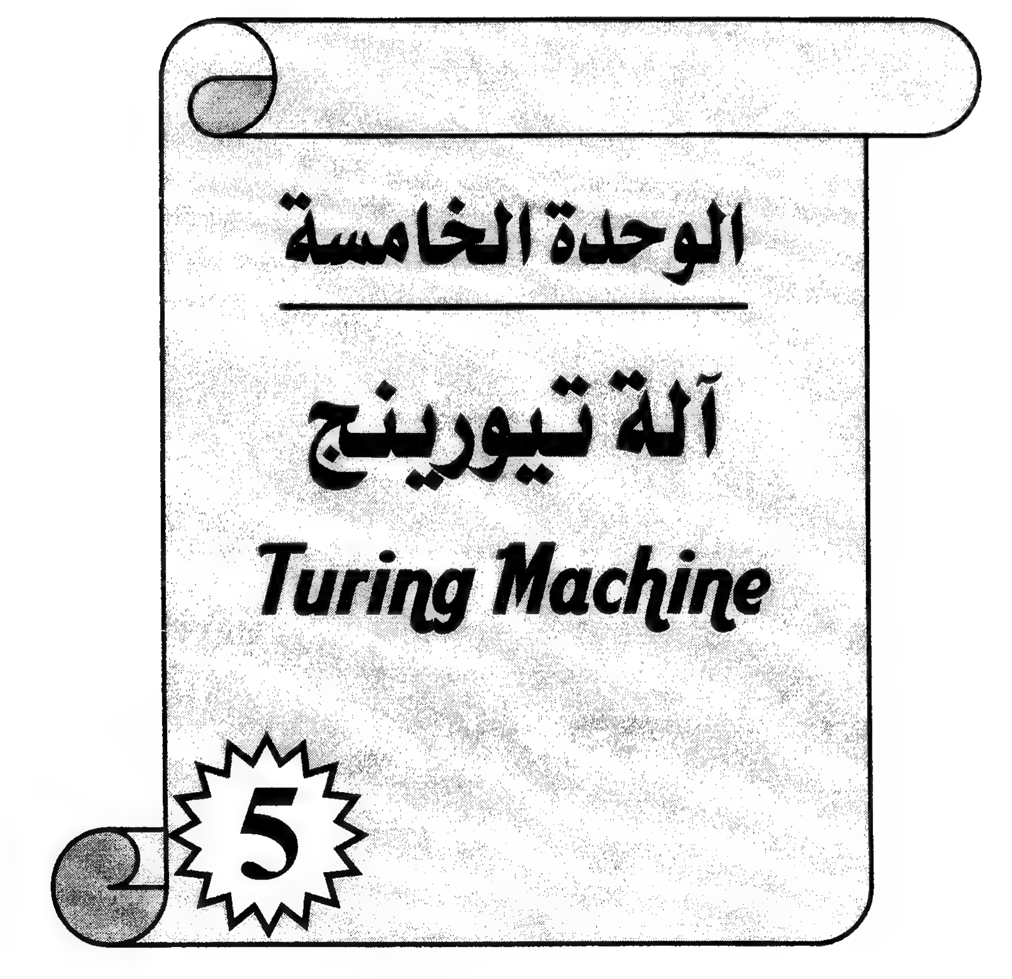
- **The PDA M Accepting L':**

Q = {q}
Σ = T = {a, b}
Γ = V = {S}
z = S

δ(q, a, S) = {(q, S)}
δ(q, b, S) = {(q, ε)}
δ(q, ε, S) = Ø

- If δ(q, ε, S) = {(q, ε)} is added then:

L(M) = {ε, a, aa, aaa, ..., b, ab, aab, aaab, ...}



الوحدة الخامسة

آلة تيورينج

Turing Machine

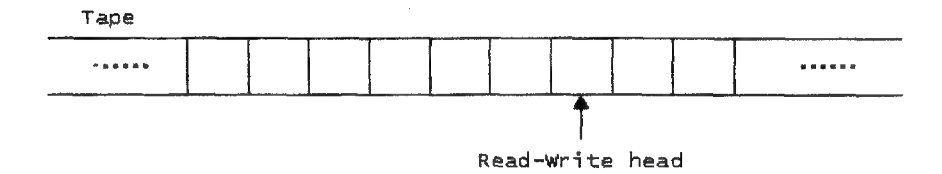
5

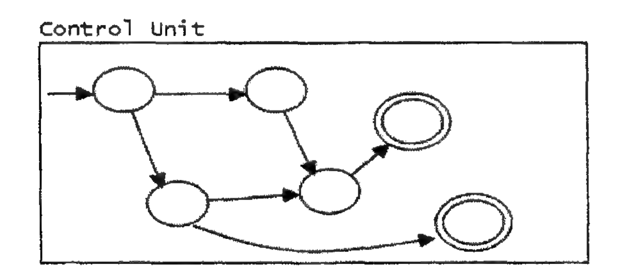
1.5 التعريف بالة تيورينج:

تعتتبر الة تيورينج نموذجا من الالات الحالة المنتهية ولهذه الالة تطبيقات كثيرة خاصة في علوم الحاسوب حيث يمكن استخدام هذه الالة في بناء معالجات النصوص وذلك لتمثيل أهم العمليات في معالجات النصوص مثل عمليات النقل والتحريك والازاحة وغيرها من العمليات والتي سنستعرض بعضا منها في هذه الوحدة ان شاء الله.

تشبه الة تيورينج الات الحالة المنتهية والتي استعرضناها سابقا في الكتاب الا انها تختلف قليلا عنها وفيما يلي سوف نستعرض أهم خصائص هذه الالة:

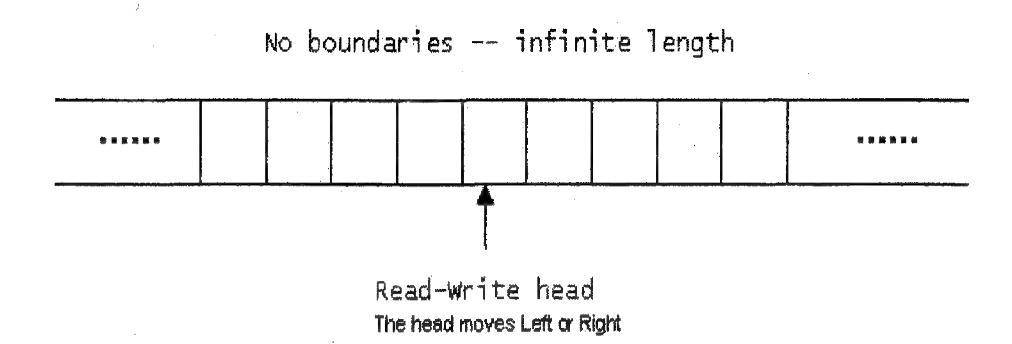
- تتكون الة تيورينج من وحدة تحكم وشريط مدخلات ورأس للقراءة والكتابة وكما هو مبين في الشمل التالي:



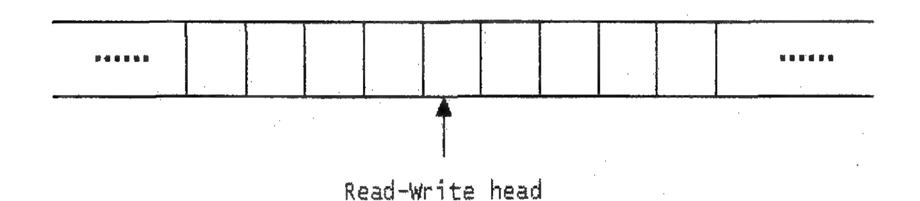


تستخدم وحدة التحكم للاحتفاظ بحالات الالة والناجمة عن عمليات القراءة أو الكتابة أو تحريك راس القراءة والكتابة لليمين أواليسار.

- سلسلة الرموز المخزنة على الشريط غير محدودة ويستطيع راس القراءة والكتابة التحرك لليسار أوالى اليمين ويكون عدد هذه الحركات غير محدود.

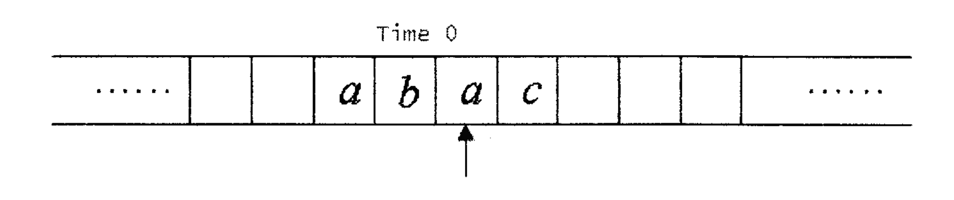


- في اللحظة الزمنية المعينة يمكن لراس القراءة والكتابة قراءة رمز أو كتابة رمز أو التحرك لليسار أو اليمين ولخطوة واحدة.



The head at each time step:
1. Reads a symbol
2. Writes a symbol
3. Moves Left or Right

- عملية القراءة أو الكتابة تتم في نفس موقع رأس القراءة والكتابة دون تحريك هذه الراس اما عماية التحريك فتتم باستخدام الرمز R للحركة لليمين والرمز H للحركة لليسار L وعليه فان مجموعة الرموز يجب ان لا تتضمن هذين الحرفين.



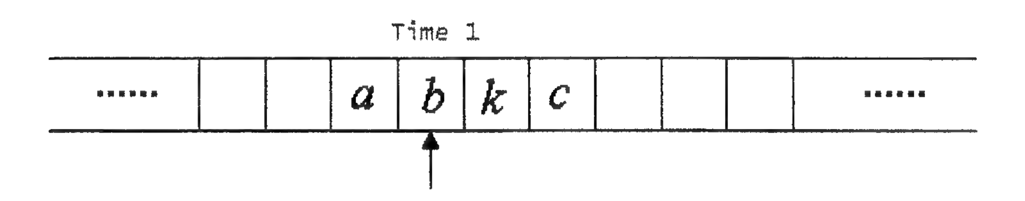
Time 1

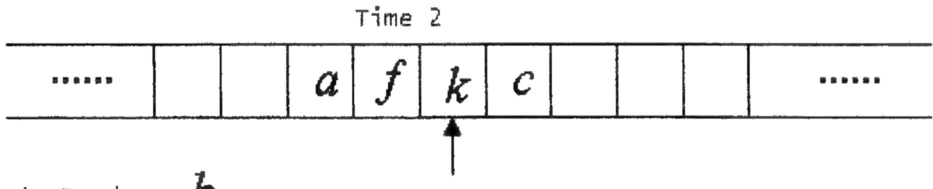
.... a b k c ....

1. Reads a

2. Writes k

3. Moves left



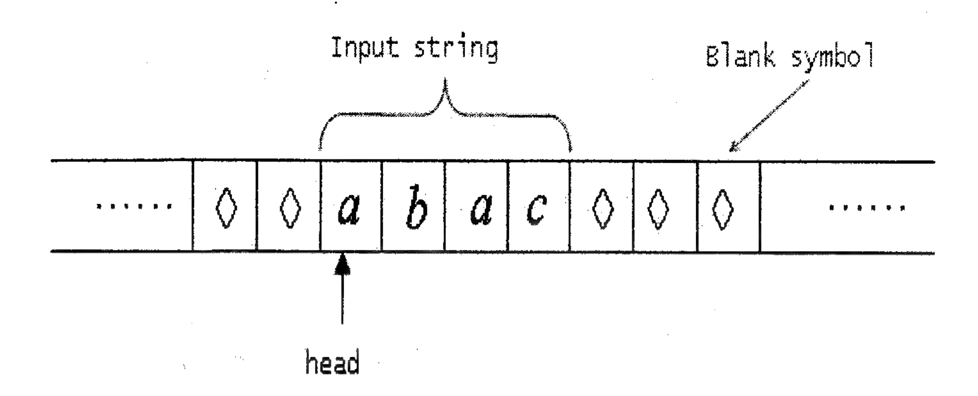


1. Reads b

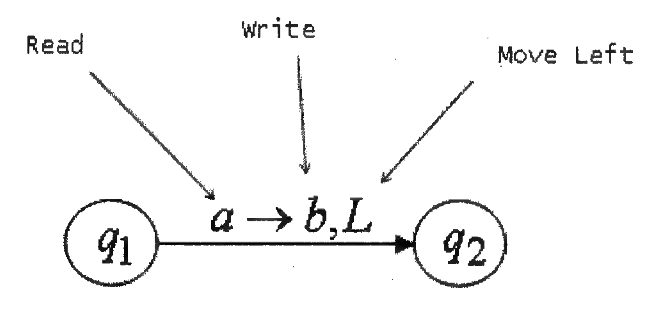
2. Writes f

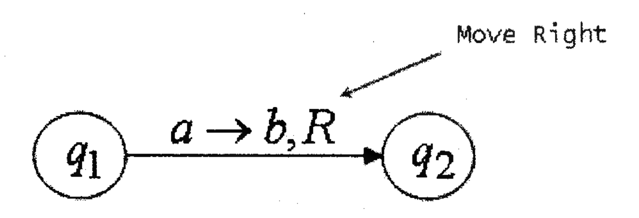
3. Moves right

- **يجـب أن تتـضمن مجموعـة الرمـوز رمـز الفـراغ وفي الغالـب يكـون الوضـع الابتدائي لرأس القراءة والكتابة عند أول فراغ من اليسار.**



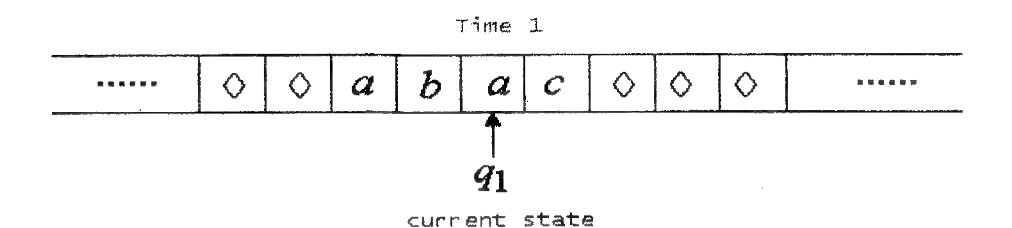
- تمثل الحالة بالدائرة وعملية الانتقال بالسهم على أن يوضع على محددات عملية الإنتقال من الحالة الحالية الى الحالة التالية وكما هو مبين في الشكل التالي:

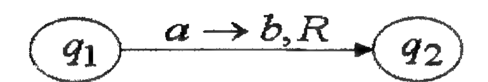




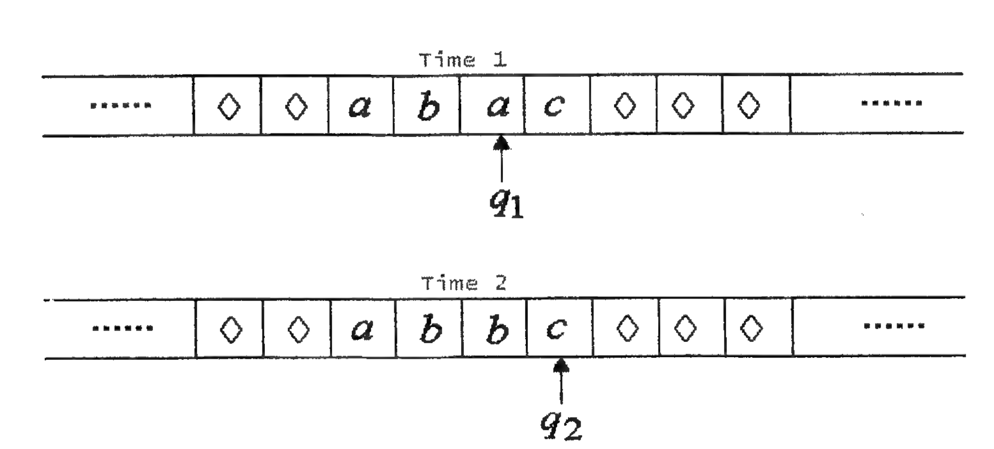
وفيما يلي بعض الاشكال التوضيحية:

.1

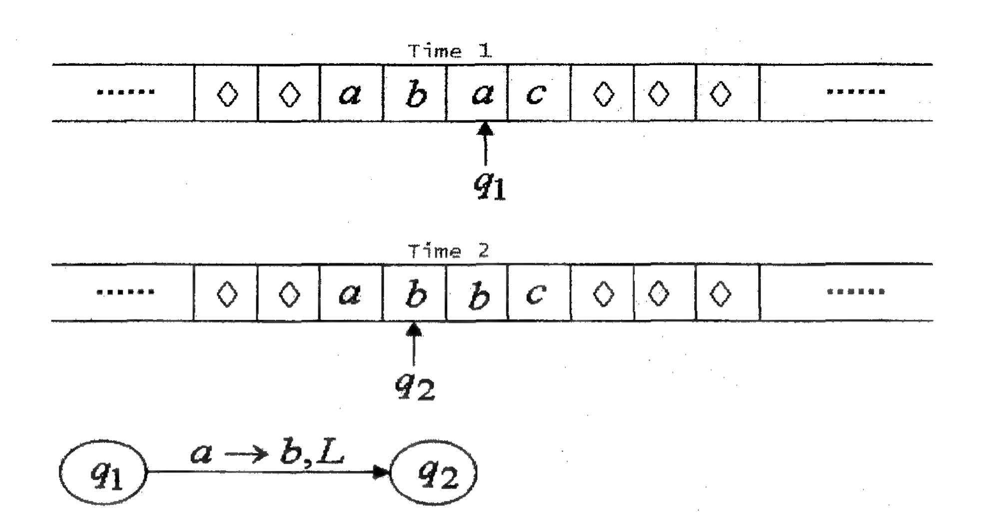




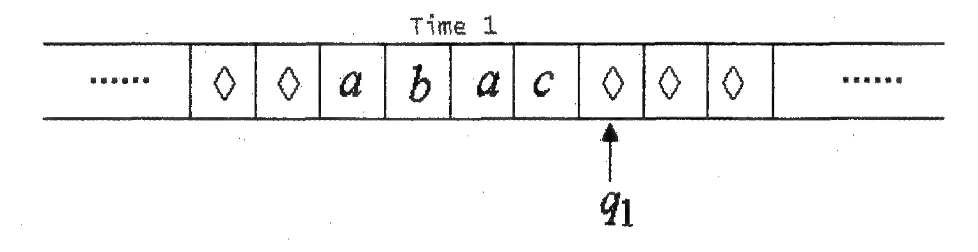
.2

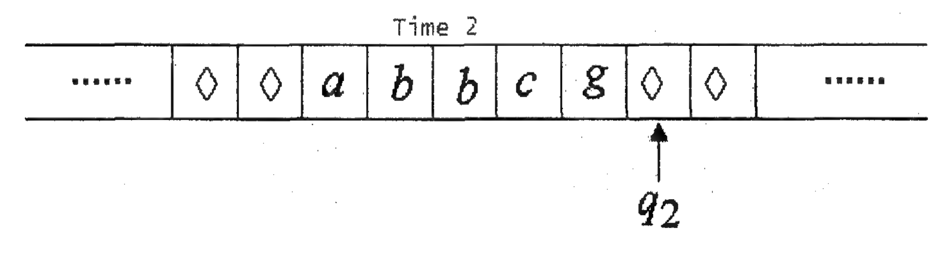


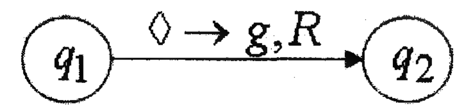
.3


Time 1
...... ◊ ◊ a b a c ◊ ◊ ◊ ......
q1
Time 2
...... ◊ ◊ a b b c ◊ ◊ ◊ ......
q2
q1
a → b,L
q2

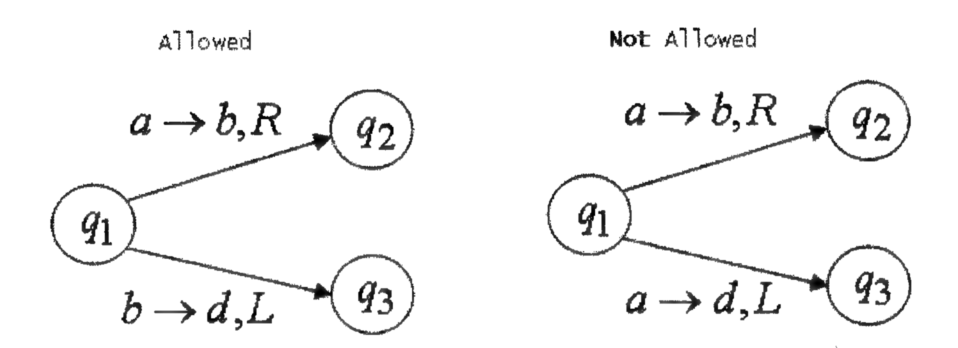
.4


Time 1
...... ◊ ◊ a b a c ◊ ◊ ◊ ......
q1

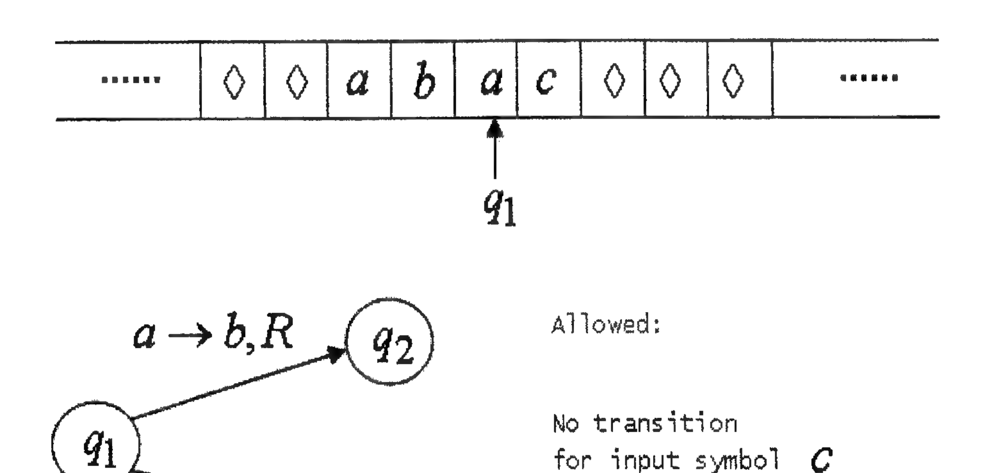

Time 2
...... ◊ ◊ a b b c g ◊ ◊ ......
q2


q1
◊ → g,R
q2

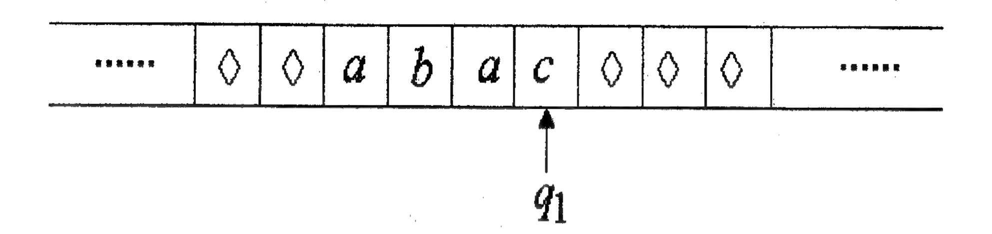
- تعتبر الة تيورينج من الالات المحدودة والتي لا تقبل الفرع في المسارات وبهذا فانها تختلف عن الة الحالة المنتهية غيرالمحدودة وتتشابه مع الة الحالة المنتهية المحدودة والشكل التالي يوضح هذه الفكرة.

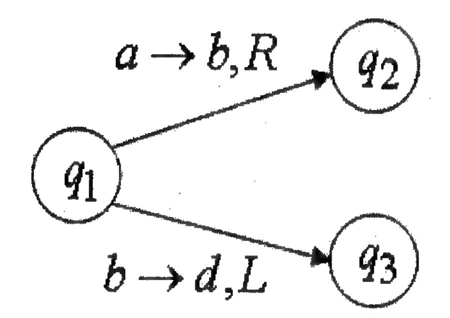


والشكل التالي يبين بعض عمليات الإنتقال المسموحة في الة تيورينج:



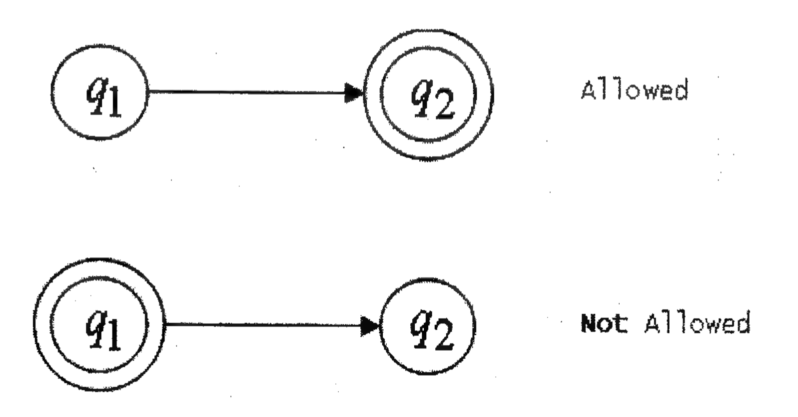
- تنتقل الة تيورينج الى حالة التوقف في حالة عدم تحقق شروط عملية النتقال من حالة لاخرى وكما هو مبين في الشكل التالي:



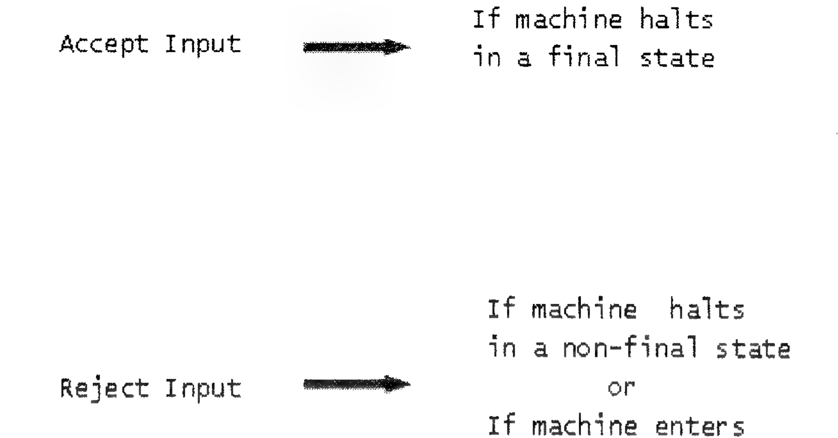


No possible transition **HALT**

- لا يجوز الإنتقال من حالة نهاشية الى حالة مقبولة من قبل الالة ويمكن النتقال من حالة مقبولة الى حالة نهائية.



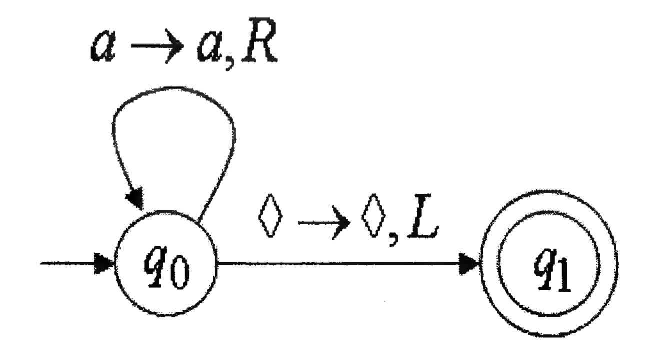
- تقبل الة تيورينج مجموعة المدخلات إذا كانت تقود الى حالة توقف نهائية وترفضها اذا كانت تقود الى حالة ليست نهائية أوحالة لا نهائية.



وفيما يلي نستعرض بعض الامثلة والتي تبين كيفية معالجة الرموز والإنتقال من حالة لاخرى:

مثال:

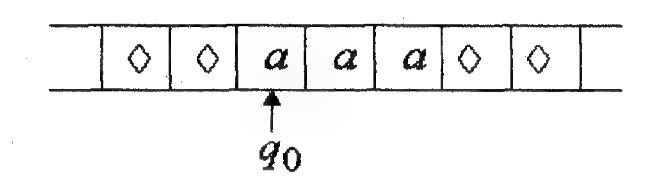
ابن الة تيورينج والتي تقبل مجموعة متتابعة من الحرف a يجب ان يبدء راس القراءة والكتابة بهذا الحرف ومن ثم تحريك الراس لليمين عند قراءة هذا الحرف والابقاء عليه:

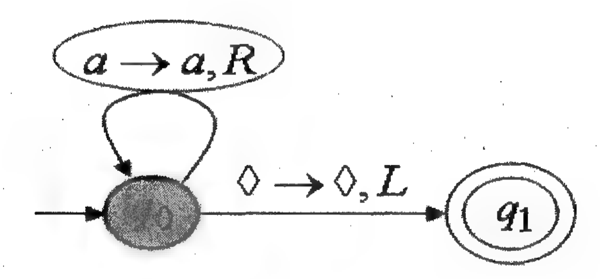


والاشكال التالية تبين تسلسل تنفيذ هذه الالة:

.1

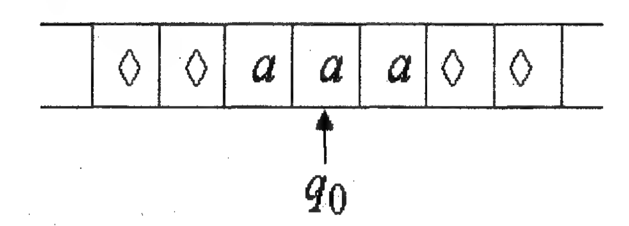
Time 0

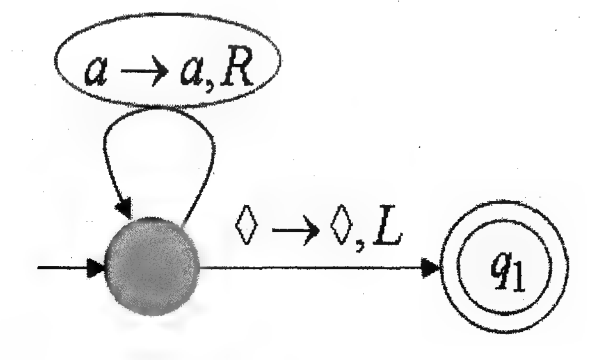




.2

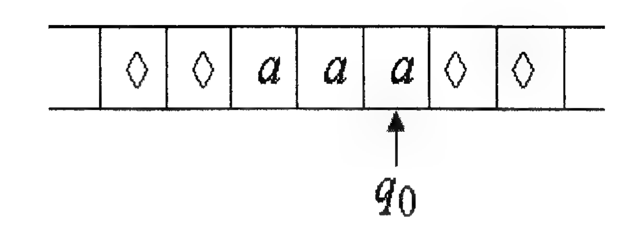
Time 1

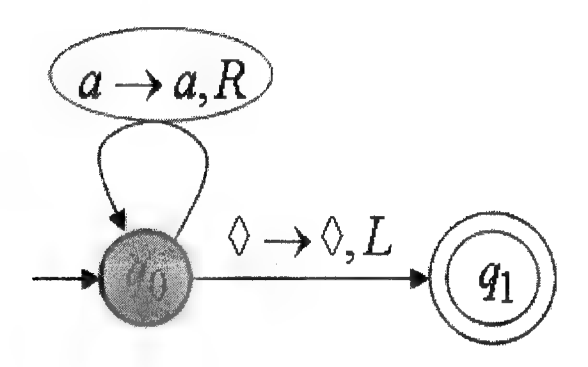




.3

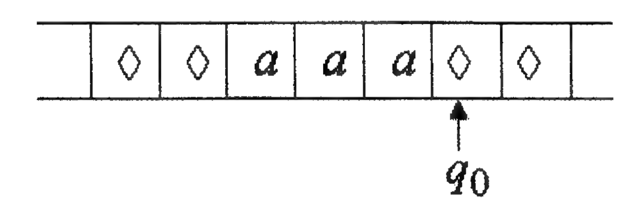
Time 2

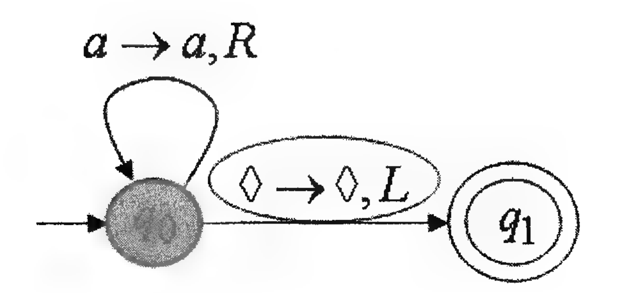




.4

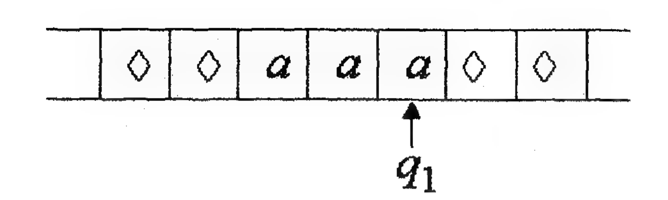
Time 3

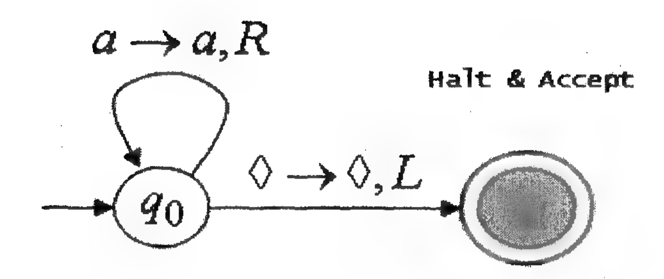




.5



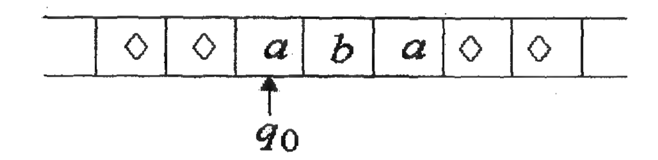


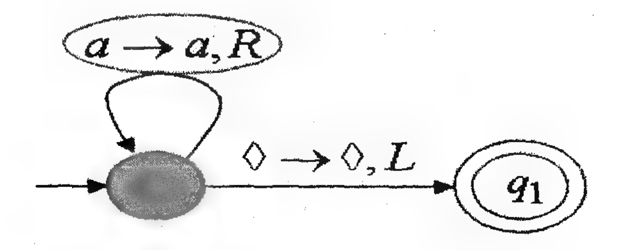


اما عملية رفض السلسلة الرمزية(نفس السلسلة في المثال السابق) فيمكن بيانها من خلال الأشكال التالية:

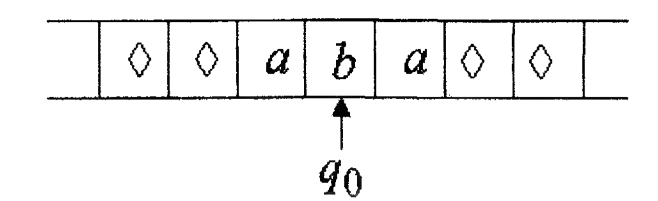
.1

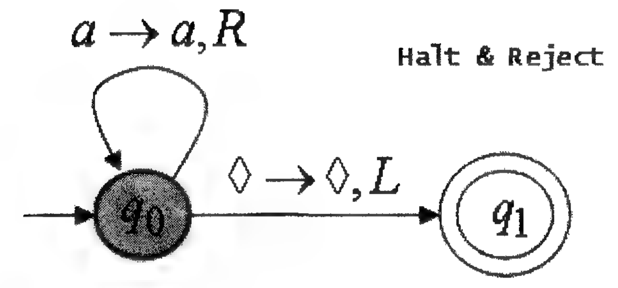
Time 0





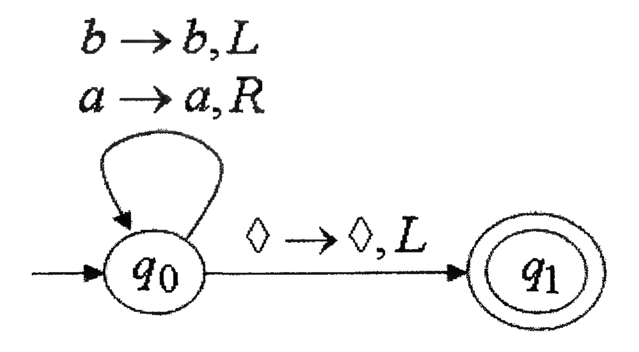
.2





مثال:

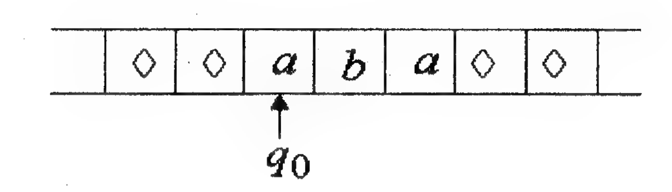
فيما يلي الة مخصصة للتعرف على مجموعة من الاحرف المتتابعة لـ a والتي سوف تنتقل إلى حالة لا نهائية نظرا لعدم توفر هذه الرموز في مجموعة الرموز الموجودة على الشريط:

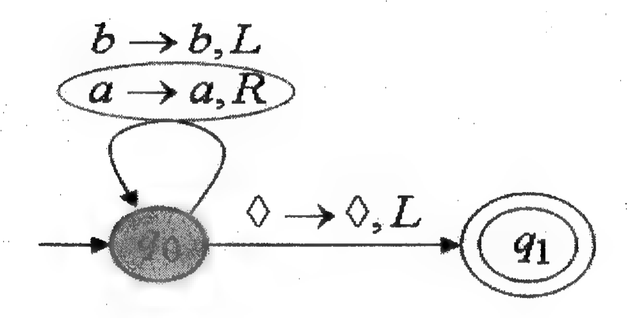


وفيما يلي الية تنفيذ هذه الالة:

.1

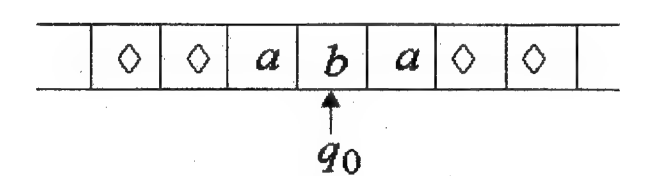
Time 0

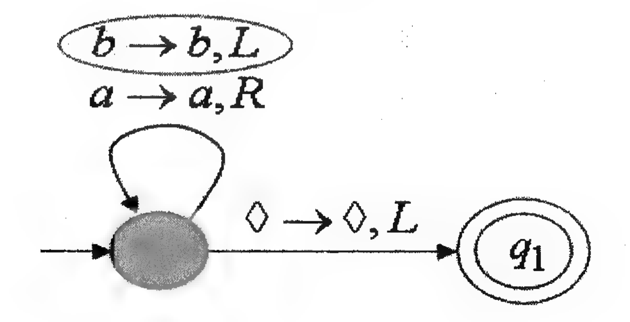




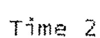
.2

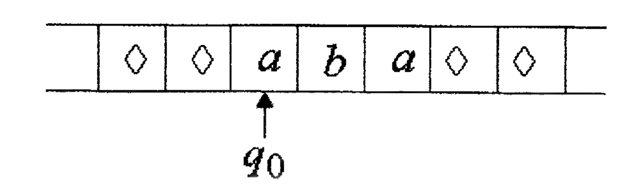
Time 1

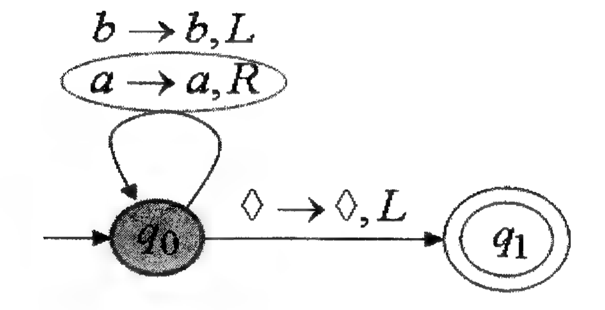




.3

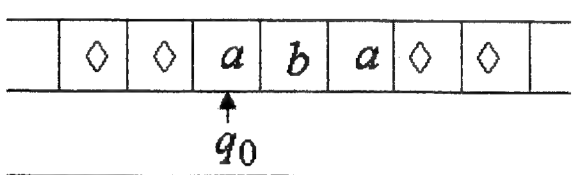




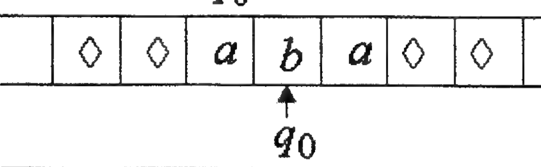


.4

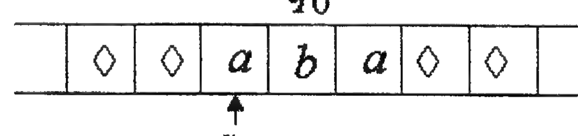
Time 2



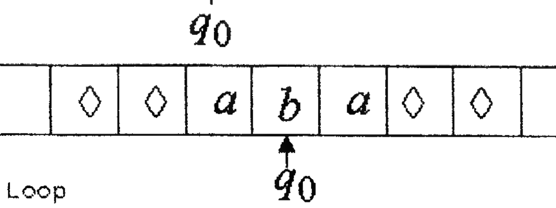
Time 3



Time 4



Time 5



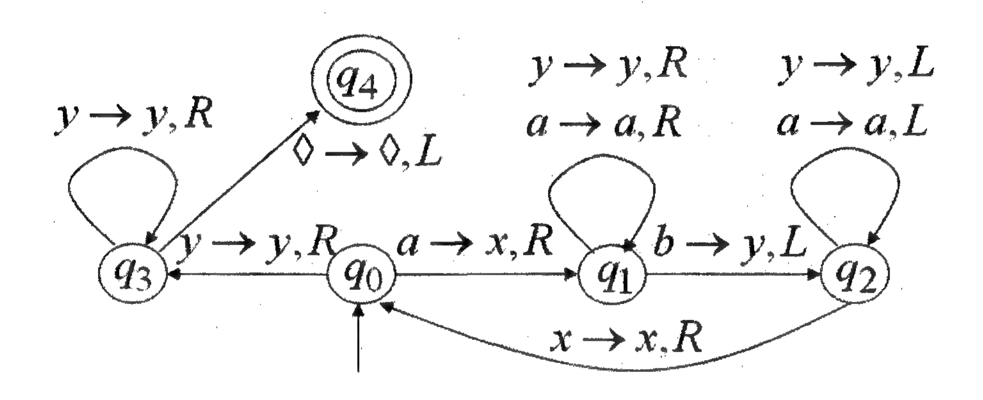
... Infinite Loop

مثال:

صمم الة تيورينج والتي تقبل التعبير أو اللغة التالية:

$\{ a^n b^n \}$

الحل:



وفيما يلي الية تنفيذ هذه الالة والهيئات التي تمر بها:

.1

Time 0

	◊	a	a	b	b	◊	◊

q_0

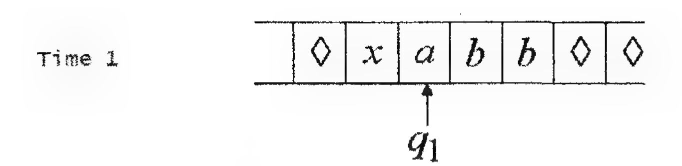
$y \rightarrow y, R$ $a \rightarrow a, R$ $y \rightarrow y, L$ $a \rightarrow a, L$

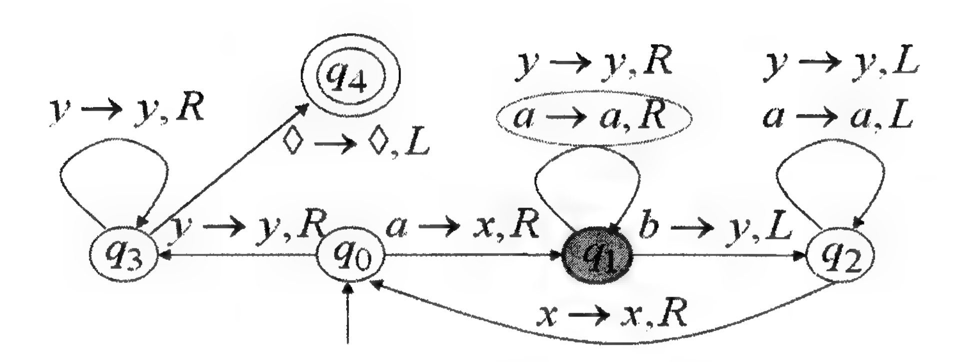
q_4 $y \rightarrow y, R$ $\Diamond \rightarrow \Diamond, L$

q_3 $y \rightarrow y, R$ q_0 $a \rightarrow x, R$ q_1 $b \rightarrow y, L$ q_2

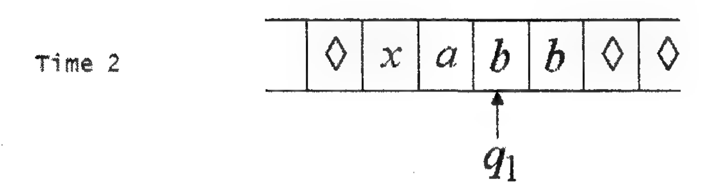
$x \rightarrow x, R$

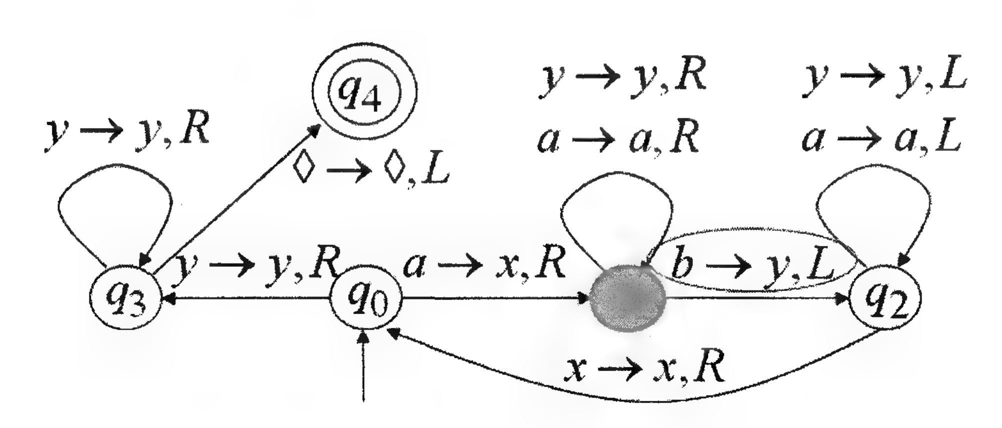
.2


Time 1
◊
x
a
b
b
◊
◊
q_1

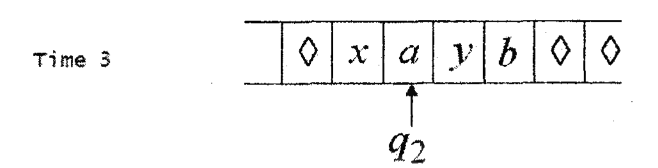

q_4
$y \to y, R$
$\lozenge \to \lozenge, L$
$y \to y, R$
$a \to a, R$
$y \to y, L$
$a \to a, L$
$y \to y, R$
$a \to x, R$
$b \to y, L$
q_3
q_0
q_1
q_2
$x \to x, R$

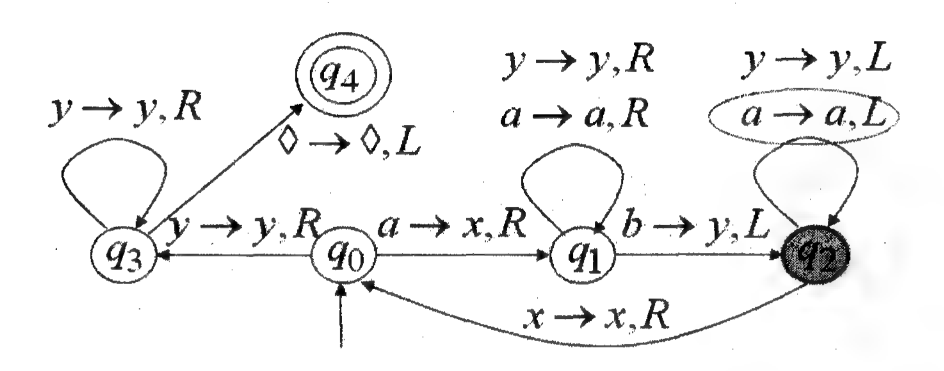
.3


Time 2
◊
x
a
b
b
◊
◊
q_1

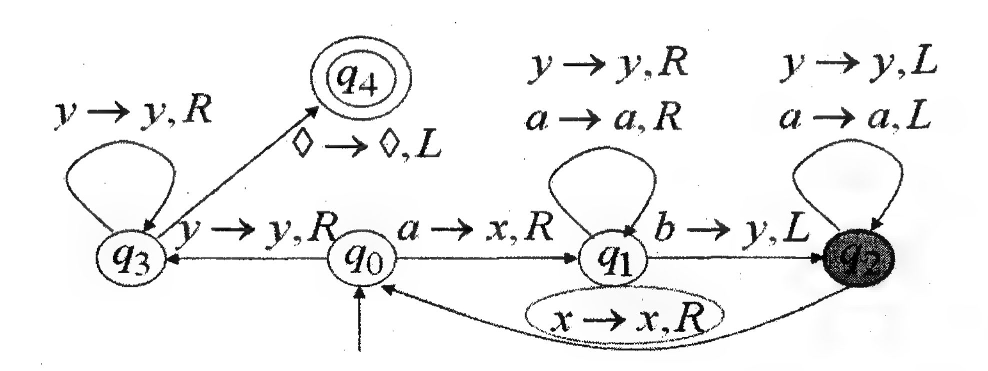

q_4
$y \to y, R$
$\lozenge \to \lozenge, L$
$y \to y, R$
$a \to a, R$
$y \to y, L$
$a \to a, L$
$y \to y, R$
$a \to x, R$
$b \to y, L$
q_3
q_0
q_2
$x \to x, R$

.4


Time 3
◊
x
a
y
b
◊
◊
q_2

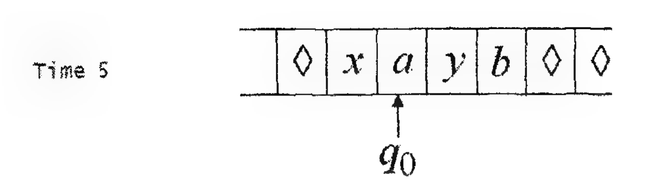

q_4
$y \rightarrow y, R$
$\Diamond \rightarrow \Diamond, L$
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
$y \rightarrow y, R$
$a \rightarrow x, R$
$b \rightarrow y, L$
q_3
q_0
q_1
q_2
$x \rightarrow x, R$

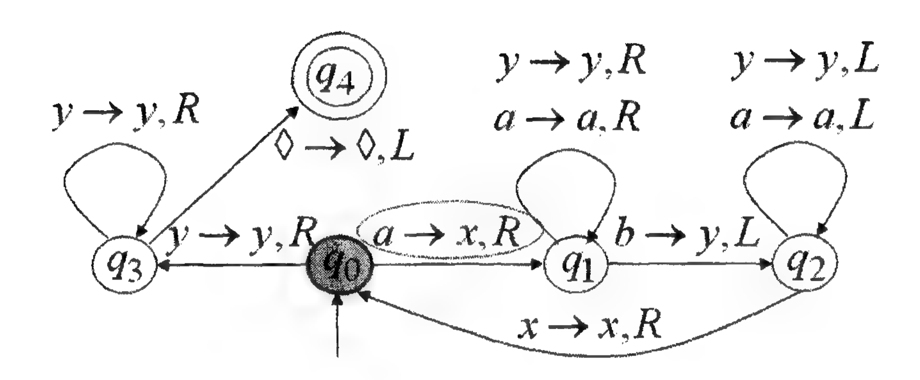
.5


Time 4
◊
x
a
y
b
◊
◊
q_2
q_4
$y \rightarrow y, R$
$\Diamond \rightarrow \Diamond, L$
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
$y \rightarrow y, R$
$a \rightarrow x, R$
$b \rightarrow y, L$
q_3
q_0
q_1
q_2
$x \rightarrow x, R$

.6

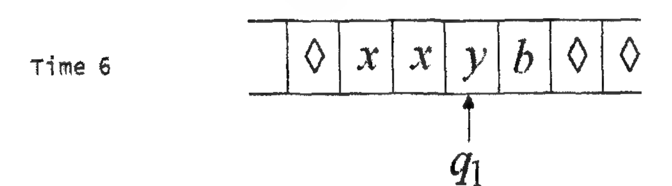
Time 5

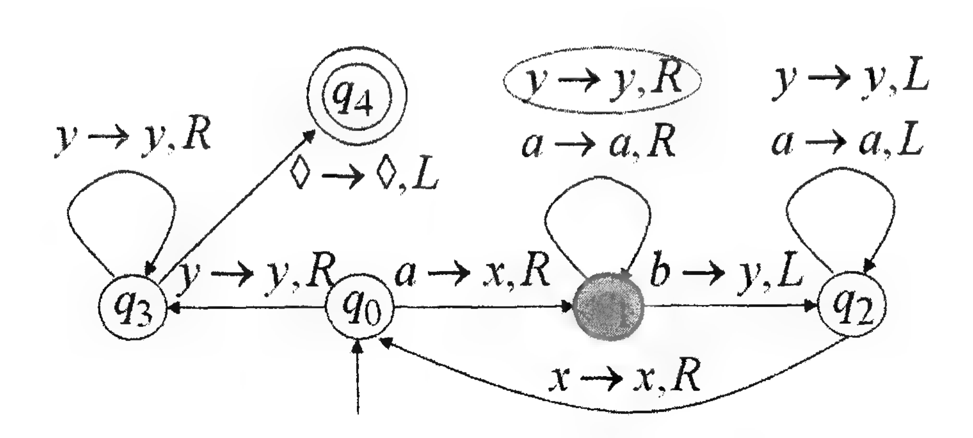




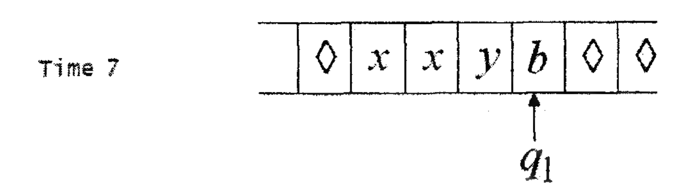
.7

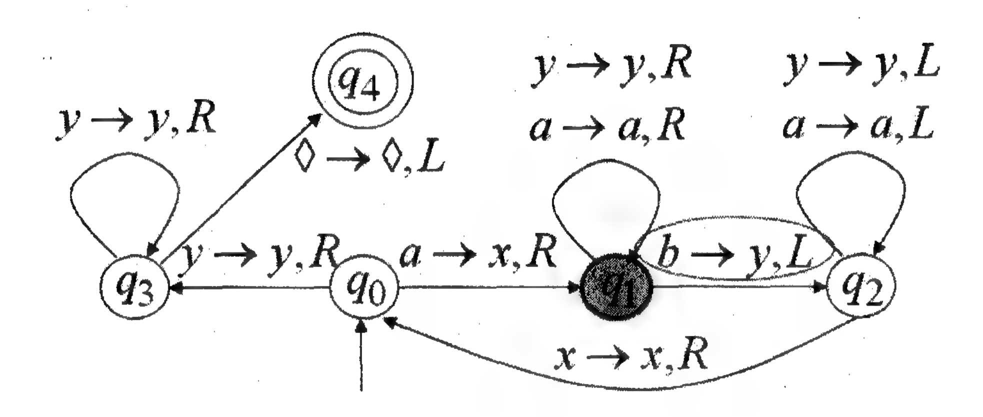
Time 6



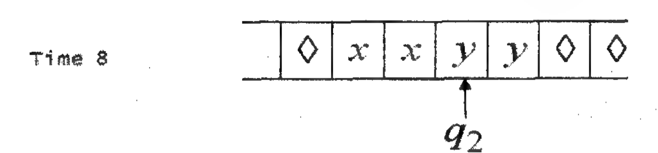


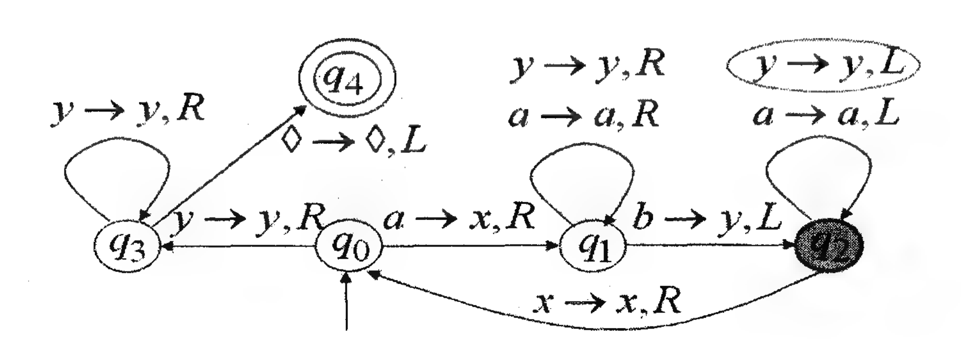
.8


Time 7
◊
x
x
y
b
◊
◊
q_1

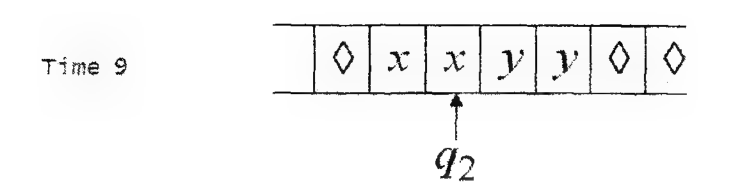

q_4
$y \to y, R$
$\Diamond \to \Diamond, L$
$y \to y, R$
$a \to a, R$
$y \to y, L$
$a \to a, L$
q_3
$y \to y, R$
q_0
$a \to x, R$
q_1
$b \to y, L$
q_2
$x \to x, R$

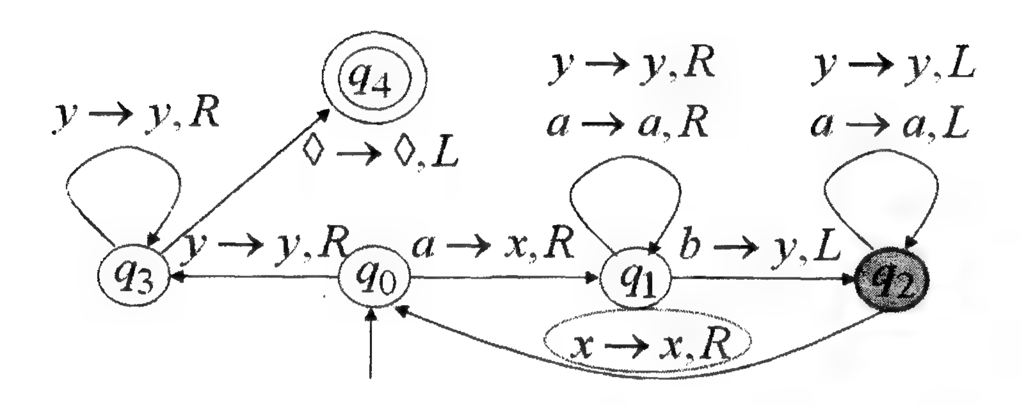
.9


Time 8
◊
x
x
y
y
◊
◊
q_2

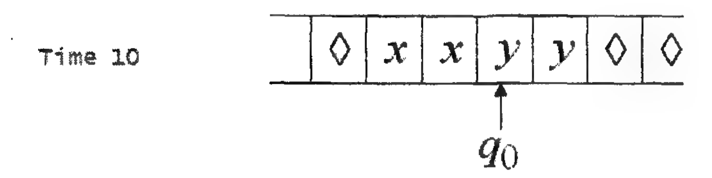

q_4
$y \to y, R$
$\Diamond \to \Diamond, L$
$y \to y, R$
$a \to a, R$
$y \to y, L$
$a \to a, L$
q_3
$y \to y, R$
q_0
$a \to x, R$
q_1
$b \to y, L$
q_2
$x \to x, R$

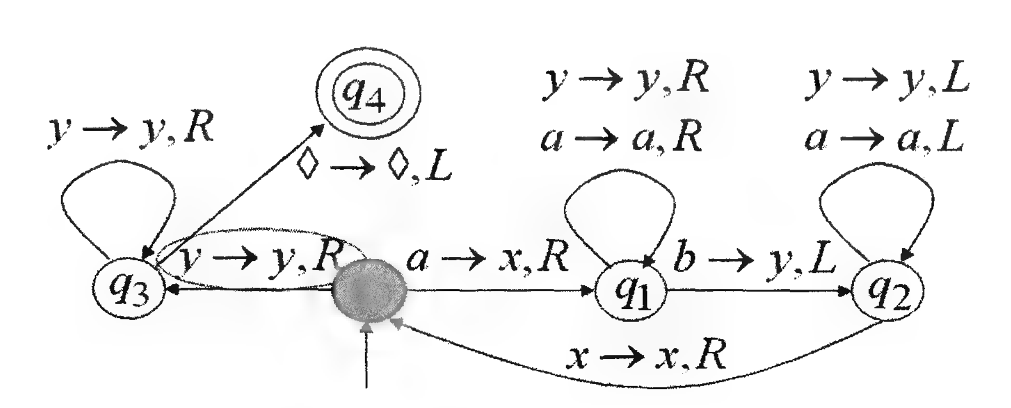
.10


Time 9
◊
x
x
y
y
◊
◊
q_2

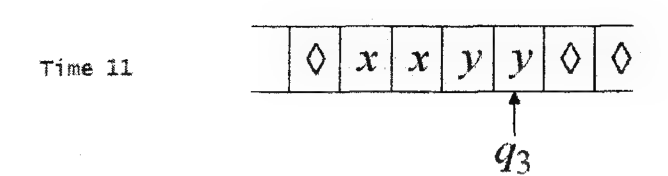

q_4
$y \to y, R$
$\Diamond \to \Diamond, L$
$y \to y, R$
$a \to a, R$
$y \to y, L$
$a \to a, L$
$y \to y, R$
$a \to x, R$
$b \to y, L$
q_3
q_0
q_1
q_2
$x \to x, R$

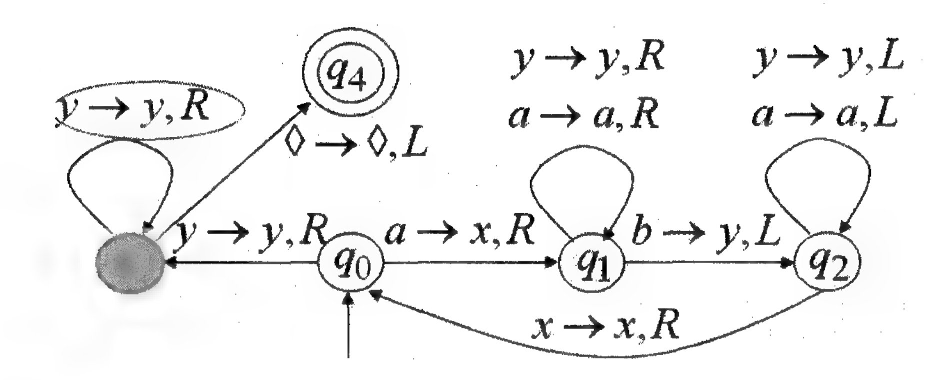
.11


Time 10
◊
x
x
y
y
◊
◊
q_0

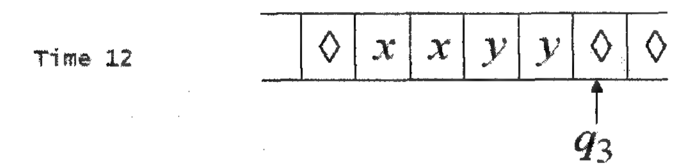

q_4
$y \to y, R$
$\Diamond \to \Diamond, L$
$y \to y, R$
$a \to a, R$
$y \to y, L$
$a \to a, L$
$y \to y, R$
$a \to x, R$
$b \to y, L$
q_3
q_1
q_2
$x \to x, R$

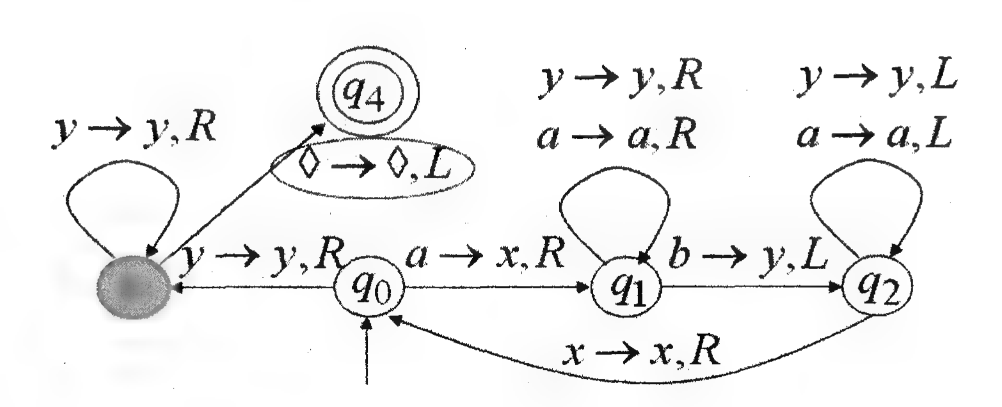
.12


Time 11
◊
x
x
y
y
◊
◊
q_3

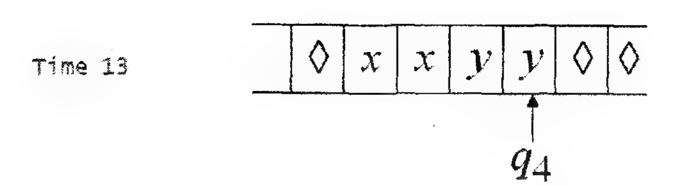

q_4
$y \rightarrow y,R$
$\Diamond \rightarrow \Diamond,L$
$y \rightarrow y,R$
$a \rightarrow a,R$
$y \rightarrow y,L$
$a \rightarrow a,L$
$y \rightarrow y,R$
$a \rightarrow x,R$
$b \rightarrow y,L$
q_0
q_1
q_2
$x \rightarrow x,R$

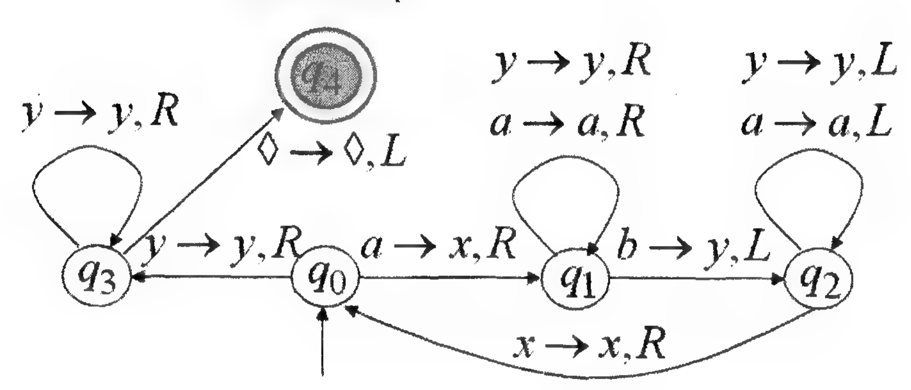
.13


Time 12
◊
x
x
y
y
◊
◊
q_3


q_4
$y \rightarrow y,R$
$\Diamond \rightarrow \Diamond,L$
$y \rightarrow y,R$
$a \rightarrow a,R$
$y \rightarrow y,L$
$a \rightarrow a,L$
$y \rightarrow y,R$
$a \rightarrow x,R$
$b \rightarrow y,L$
q_0
q_1
q_2
$x \rightarrow x,R$

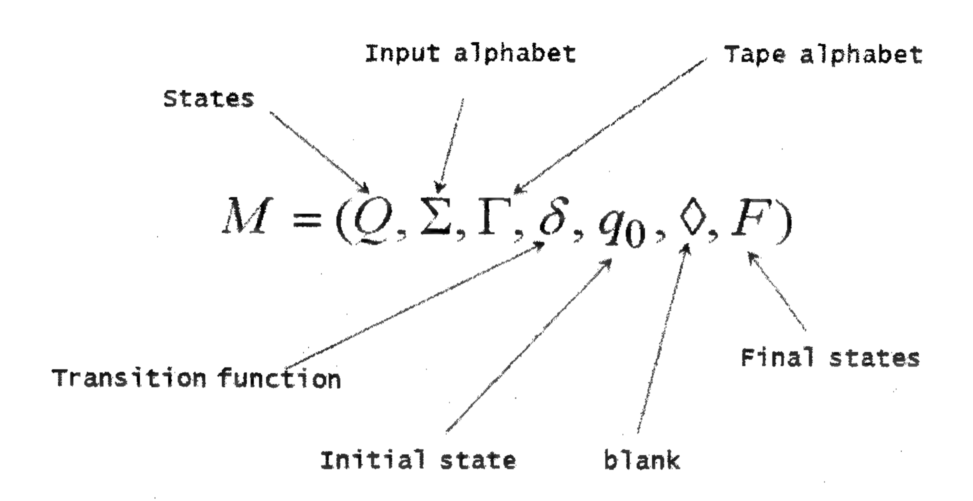
.14


Time 13
◊ x x y y ◊ ◊
q_4


Halt & Accept
q_4
$y \to y, R$
$\Diamond \to \Diamond, L$
$y \to y, R$
$a \to a, R$
$y \to y, L$
$a \to a, L$
q_3
$y \to y, R$
q_0
$a \to x, R$
q_1
$b \to y, L$
q_2
$x \to x, R$

2.5 النموذج الرياضي لالة تيورينج:

تعرف الة تيورينج رياضيا كما يلي:



حيث يضم هذا النموذج ما يلي:

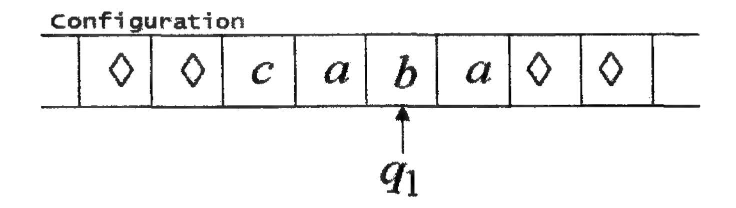
- مجموعة الحالات بما فيها الحالة الابتدائية والحالات النهائية المقبولة والحالات غيرالمقبولة والحالات المرفوضة.
- مجموعة رموز المدخلات والتي تحتوي على الفراغ ويجب ان لا تحتوي على الحرف R و L لاستخدامها لغاية تحريك رأس القراءة والكتابة لليمين أو لليسار.
- مجموعة رموز الشريط.
- الحالة الابتدائية.
- رمز الفراغ.
- مجموعة الحالات النهائية.

تستخدم دالة الانتقال للتعبير عن الحالة التي يمكن الانتقال اليها من حالة محددة بقراءة رمز للانتقال من الحالة الحالية الى الحالة القدمة مع تنفيذ عملية الكتابة على الشريط وفي نفس الموقع ثم تحريك رأس القراءة والكتابة الى اليمين أو اليسار وكما هو مبين في الشكل التالي:

$$q_1 \xrightarrow{c \rightarrow d, L} q_2$$

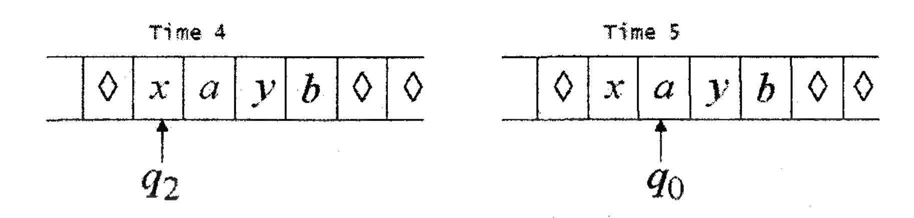
$$\delta(q_1, c) = (q_2, d, L)$$

يمكن وصف الالة ايضا بالهيئة والتي تبين الحالة الحالية ومجموعة الرموز التي قرات ومجموعة الرموز المتبقية وكما هو مبين في الشكل التالي:



Instantaneous description: $ca\, q_1\, ba$

ويمكن استخدام الهيئة لوصف سلوك الالة كما يلي:



A Move: $q_2\ xayb \succ x\ q_0\ ayb$

مثال:

For $\Sigma = \{a,b\}$, design a Turing Machine that accepts $L=\{a^n b^n : n \geq 1\}$

الحل:

$Q = \{q_0, q1, q2, q3, q4\}$,

$\Sigma = \{a,b\}$

$\Gamma = \{a,b,x,y,B\}$,

q_4: accept state

مدخلات وناتج تنفيذ الالة:

aaabbb
xaaybb

فيما يلي دالة الإنتقال وهيئات الالة:

$\delta(q_0,a)=(q_1,x,R)$; $\delta(q_1,a)=(q_1,a,R)$

$\delta(q_1,y)=(q_1,y,R)$; $\delta(q_1,b)=(q_2,y,L)$

$\delta(q_2,y)=(q_2,y,L)$; $\delta(q_2,a)=(q_2,a,L)$

$\delta(q_2,x)=(q_0,x,R)$

$\delta(q_0,y)=(q_3,y,R)$; $\delta(q_3,y)=(q_3,y,R)$

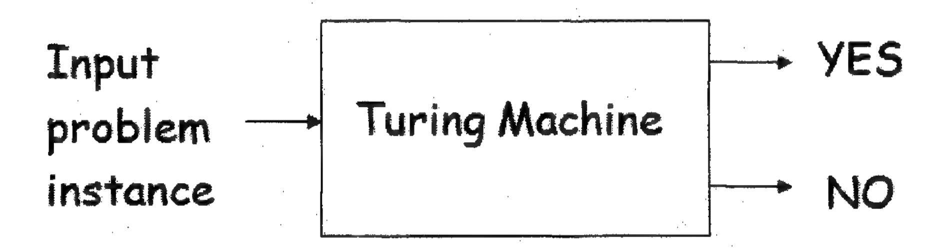
$\delta(q_3,B)=(q_4,B,R)$

$$q_0aabb \vdash xq_1abb \vdash xaq_1bb \vdash xq_2ayb \vdash q_2xayb$$
$$\vdash xq_0ayb \vdash xxq_1yb \vdash xxyq_1b \vdash xxq_2yy$$
$$\vdash xq_2xyy \vdash xxq_0yy \vdash xxyq_3y \vdash xxyyq_3B$$
$$\vdash xxyyBq_4$$

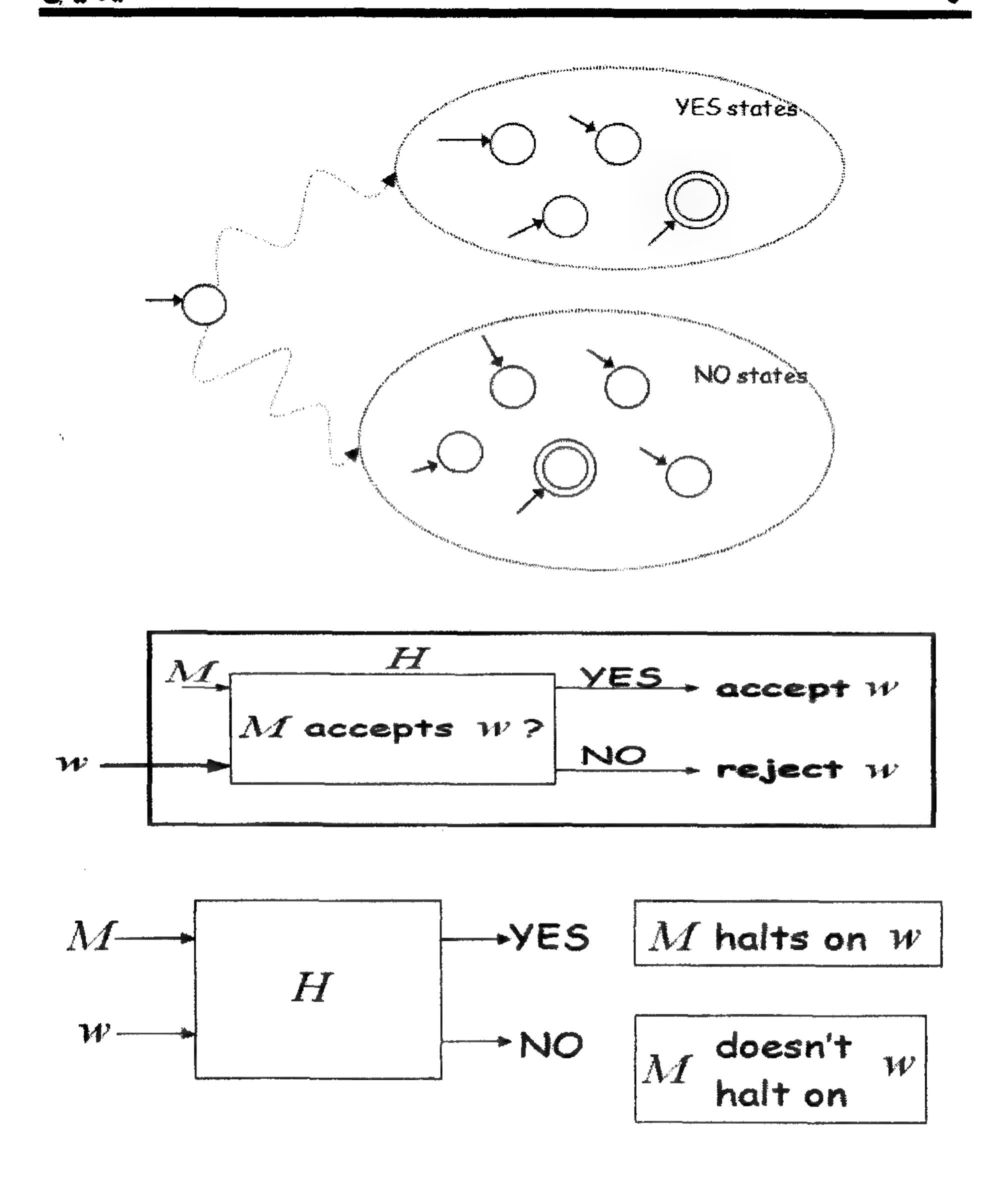
3.5 تطبيقات الة تيورينج:

تستخدم الة تيورينج في الكثير من التطبيقات ولكننا هنا سنستعرض بعضا من هذه التطبيقات وخاصة حساب الاقتران وتمثيل بعض العمليات في معالج النصوص مثل عملية النسخ والتحريك والازاحة.

كما اشرنا سابقا فان الة تيورينج تعمل على استعراض لغة أو تعبير منتظم وتقرر قبول أو رفض هذه اللغة بناء على التصميم المحدد لهذه الالة وبناء على الحالة النهائية التي تصل اليها الالة بعد معالجة الرموز فإذا كانت الحالة منتهية فان مجموعة الرموز مقبولة إما اذا وقعت الالة في حالة غيرمتنهية فان مجموعة هذه الرموز تكون غيرمقبولة أومرفوضة وبمعنى آخر يمكن تصور الة تيورينج بمسارين مسار القبول ومسار الرفض للتعبير أواللغة وكما هو مبين في الشكل التالي:

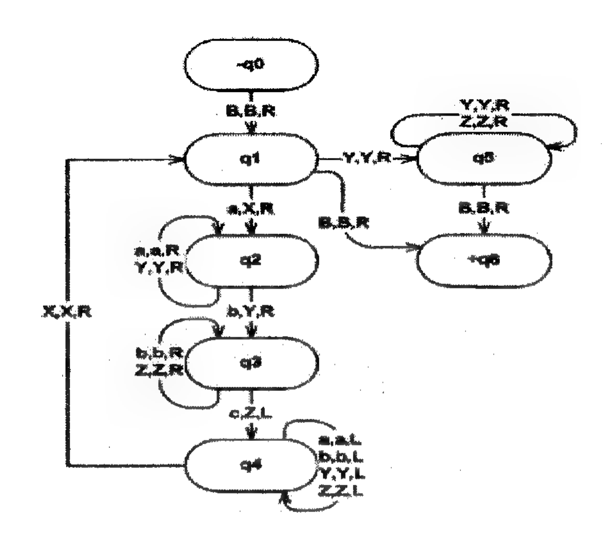


فاذا كان ناتج المسار نعم فان الالة تتوقف في حالة نهائية إما اذا كان ناتج مسار التنفيذ لا فا الالة ان تتوقف ولن تصل الى حالة نهائية وكما هو مبين في الشكل التالي:



إذا توقفت الة تيورينج في حالة نهائية فان الالة تكون محسوبة إما اذا سلكت الالة سلوك رفض الرموز فانه لن تصل الى حالة نهائية ولن تتوقف وتكون الالة في هذه الحالة غير محسوبة.

لناخذ الة تيورينج التالية:



لنتتبع هذه الالة باستخدام اللغة المؤلفة من مجموعة الرموز التالية

أولنتاكد من امكانية تمييز أو قبول هذه الرموز من قبل هذه الالة: aabbcc

فيما يلي نتيجة تتبع هذه الالة خطوة خطوة:

BaabbccB
BaabbccB
BXabbccB
BXabbccB
BXaYbccB
BXaYbccB
BXaYbZcB
BXaYbZcB
BXaYbZcB
BXaYbZcB
BXaYbZcB
BXXYbZcB
BXXYbZcB

BXXYYZcB
BXXYYZcB
BXXYYZZB
BXXYYZZB
BXXYYZZB
BXXYYZZB
BXXYYZZB
BXXYYZZB
BXXYYZZB
BXXYYZZB
BXXYYZZB
BXXYYZZBB (halt)

بما ان الالة وصلت الى حالة توقف نهائية فان هذه يعني ان مجموعة الرموز قد قبلت وتم التعرف عليها من قبل هذه الالة.

والان لنستعرض استخدام الة تيورينج في معالجة الاقترانات.

سوف نستخدم في الاقترانات القيم الاحادية بدلا من استخدام القيم العشرية أو الثنائية :

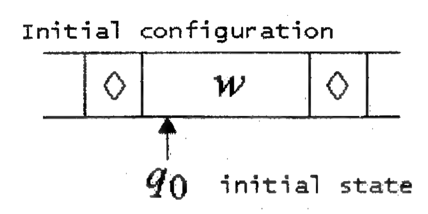
```
Decimal:    5

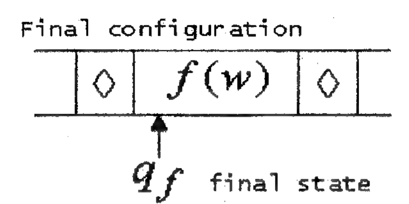
Binary:     101

Unary:      11111
```

يعتبر الاقتران محسوبا اذا توفرت الة تيورينج واستطاعت حساب هذا الاقتران أو بمعنى اخر حققت الشروط المبينة في الشكل التالي:

A function f is computable if
there is a Turing Machine M such that:

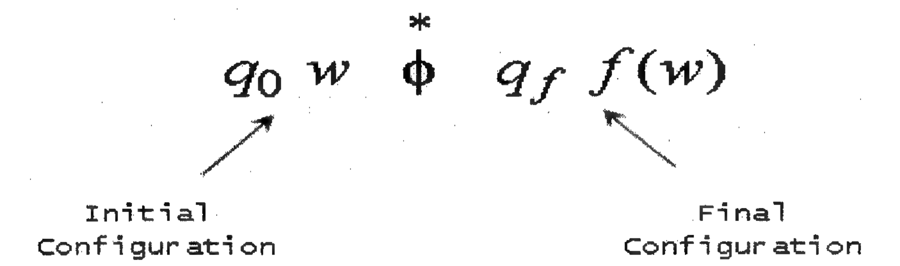




For all $w \in D$ Domain

أو

A function f is computable if
there is a Turing Machine M such that:



For all $w \in D$ Domain

والان لنأخذ الإقتران التالي:

The function $f(x, y) = x + y$ is computable

x, y are integers

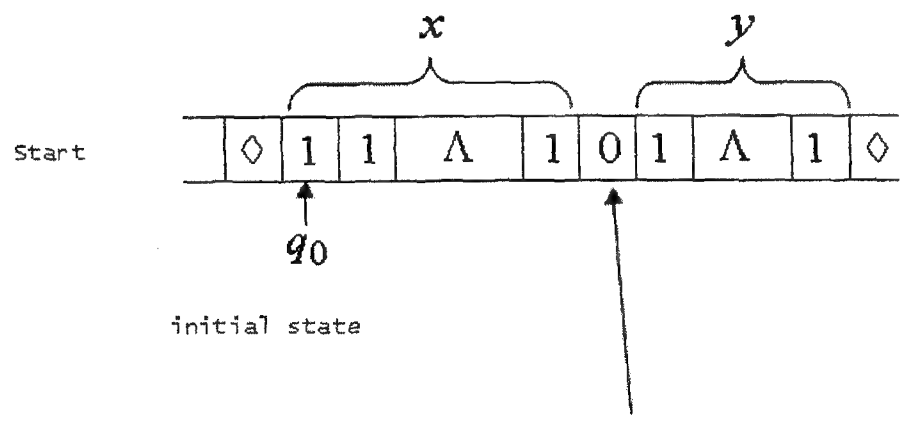
Turing Machine:

Input string: $x0y$ unary

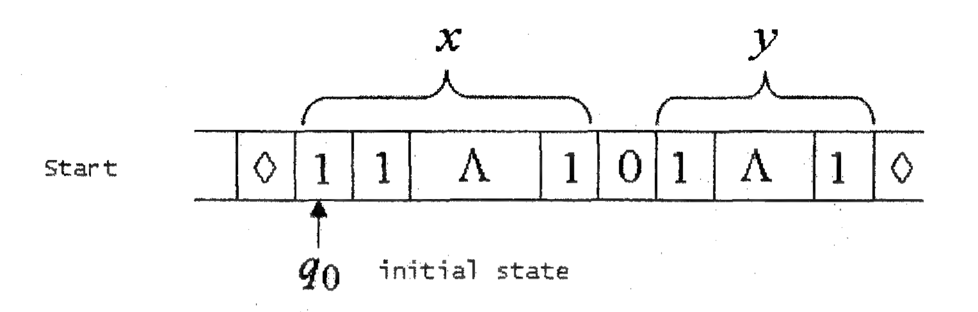
Output string: $xy0$ unary

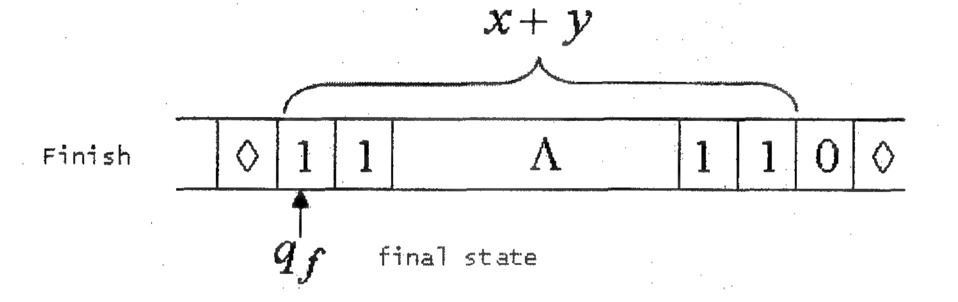
لنبدء بتمثيل الاقتران باستخدام الة تيورينج:

.1



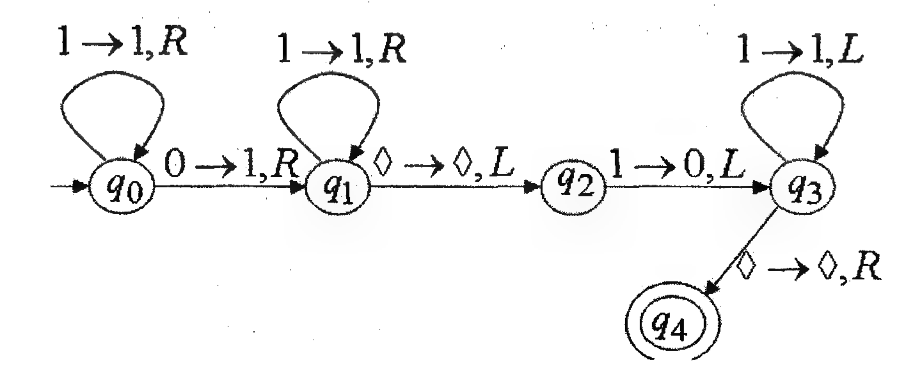
.2





وفيما يلي الة تيورينج للتعرف على الإقتران السابق:

$$f(x,y)=x+y$$



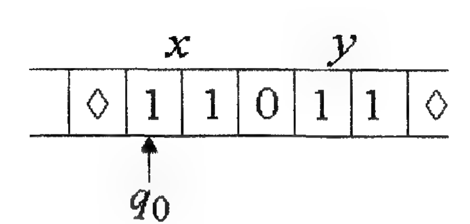
وفيما يلي ناتج تنفيذ هذه الالة:

Execution Example:

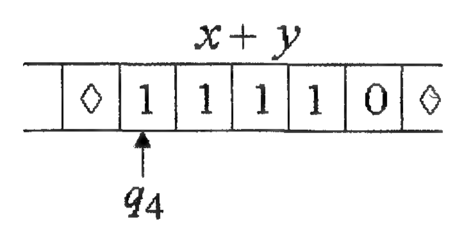


$y = 11$ (2)

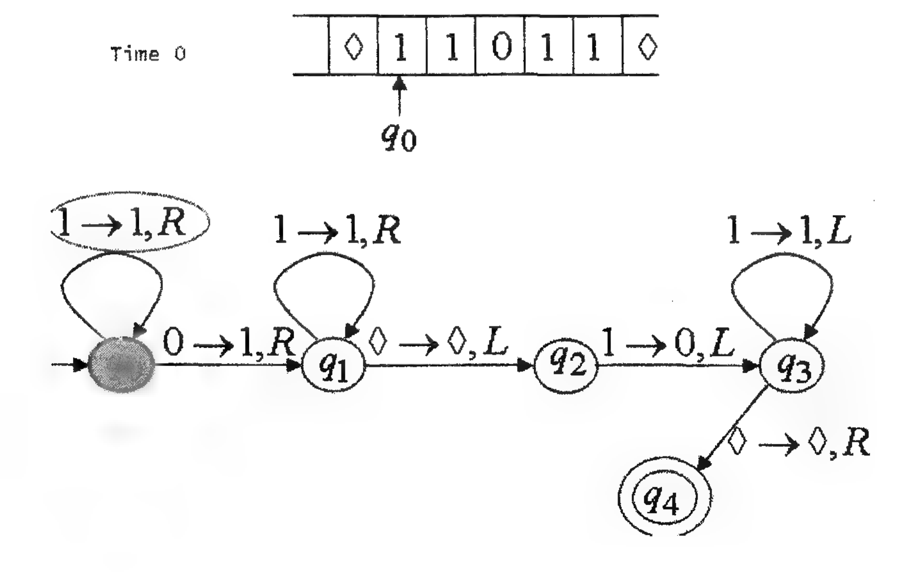
Time 0



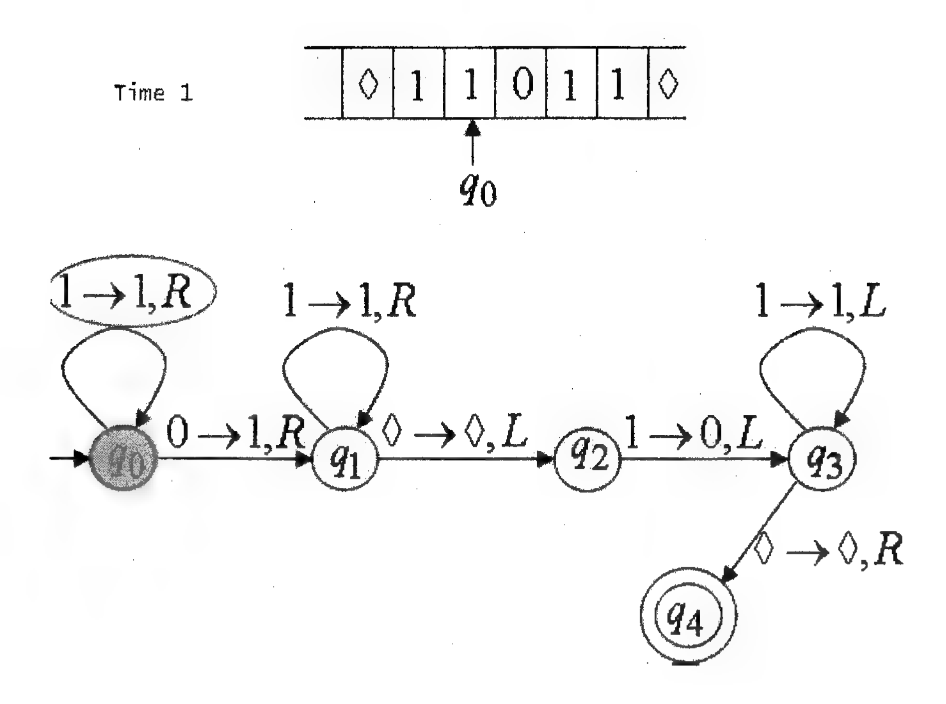
Final Result



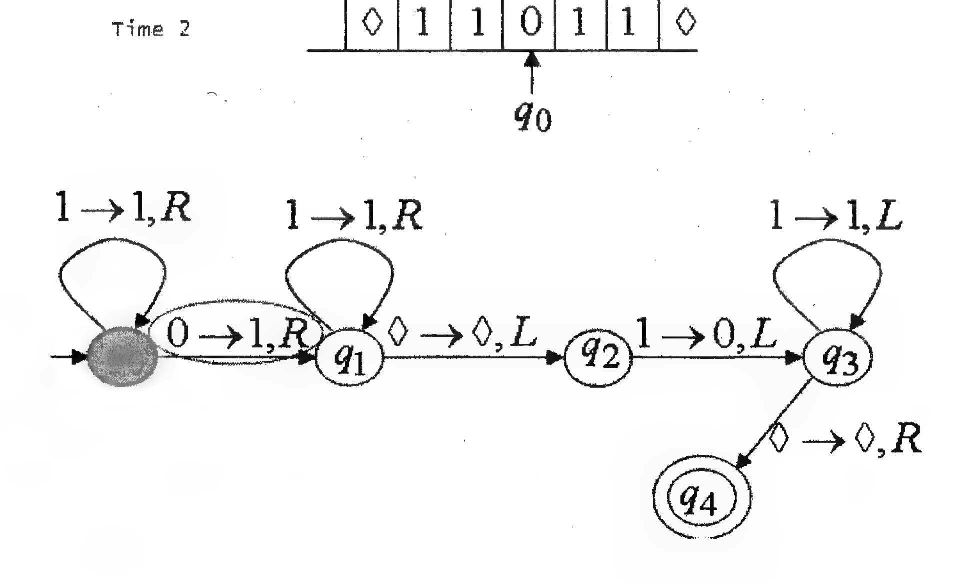
.1



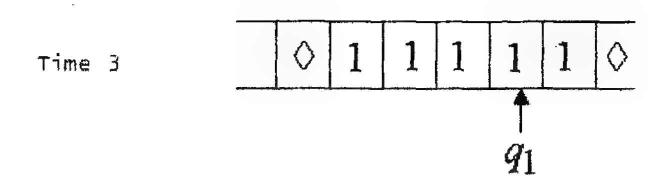
.2

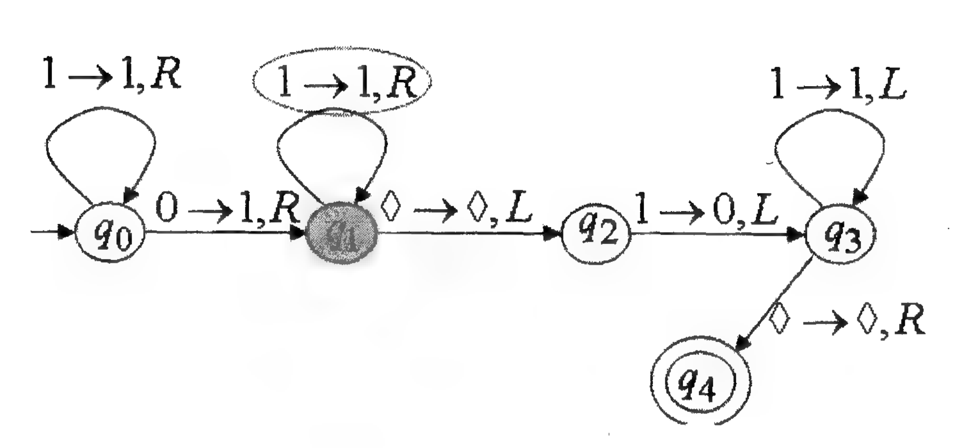

Time 1
◊ 1 1 0 1 1 ◊
q_0
$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
$0 \rightarrow 1, R$
$◊ \rightarrow ◊, L$
$1 \rightarrow 0, L$
q_0
q_1
q_2
q_3
$◊ \rightarrow ◊, R$
q_4

.3

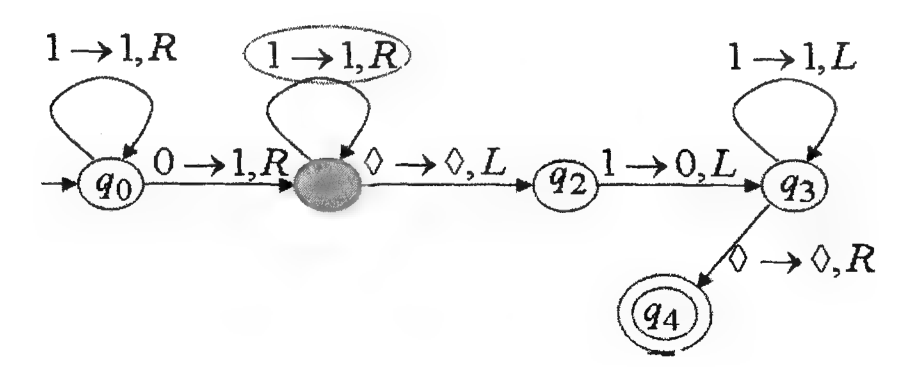

Time 2
◊ 1 1 0 1 1 ◊
q_0
$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
$0 \rightarrow 1, R$
$◊ \rightarrow ◊, L$
$1 \rightarrow 0, L$
q_1
q_2
q_3
$◊ \rightarrow ◊, R$
q_4

.4

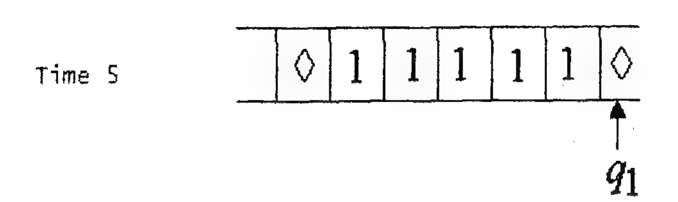

Time 3
◊
1
1
1
1
1
◊
q_1

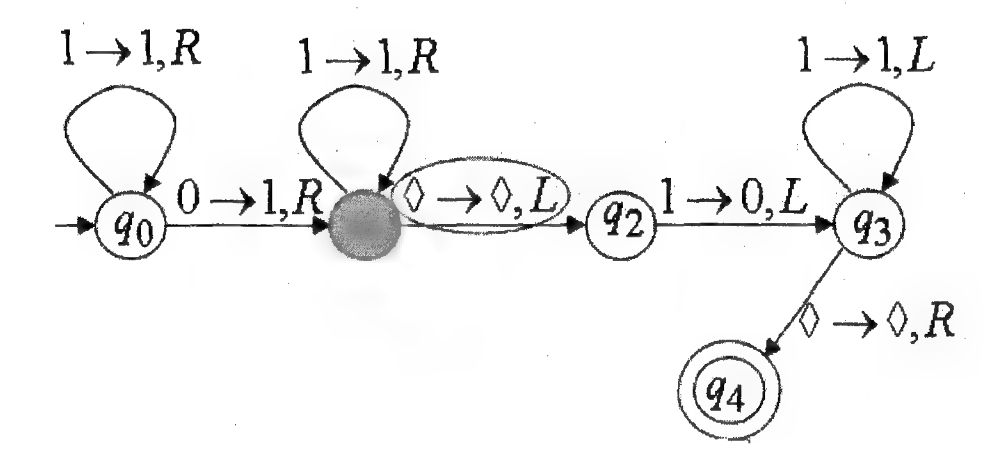

$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
$0 \rightarrow 1, R$
$\lozenge \rightarrow \lozenge, L$
$1 \rightarrow 0, L$
q_0
q_1
q_2
q_3
$\lozenge \rightarrow \lozenge, R$
q_4

.5

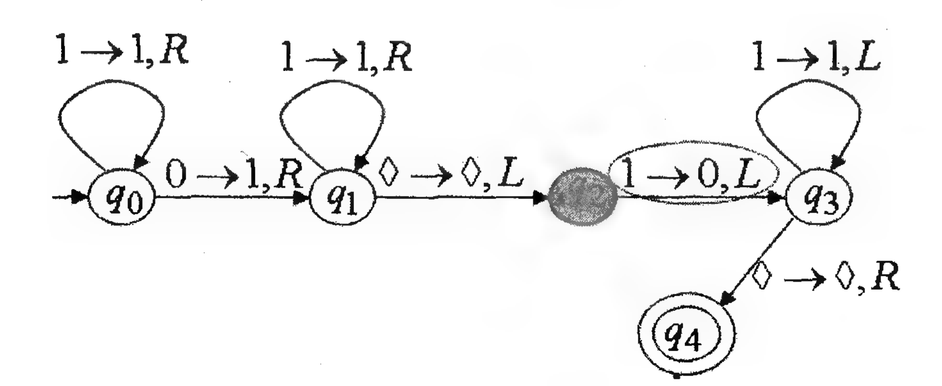

Time 4
◊
1
1
1
1
1
◊
q_1
$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
$0 \rightarrow 1, R$
$\lozenge \rightarrow \lozenge, L$
$1 \rightarrow 0, L$
q_0
q_2
q_3
$\lozenge \rightarrow \lozenge, R$
q_4

.6

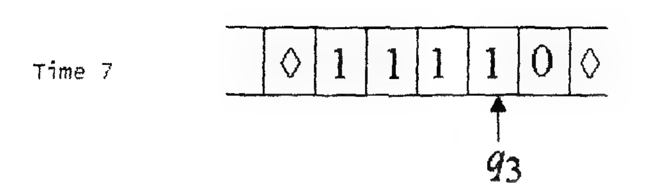

Time 5
◊
1
1
1
1
1
◊
q_1

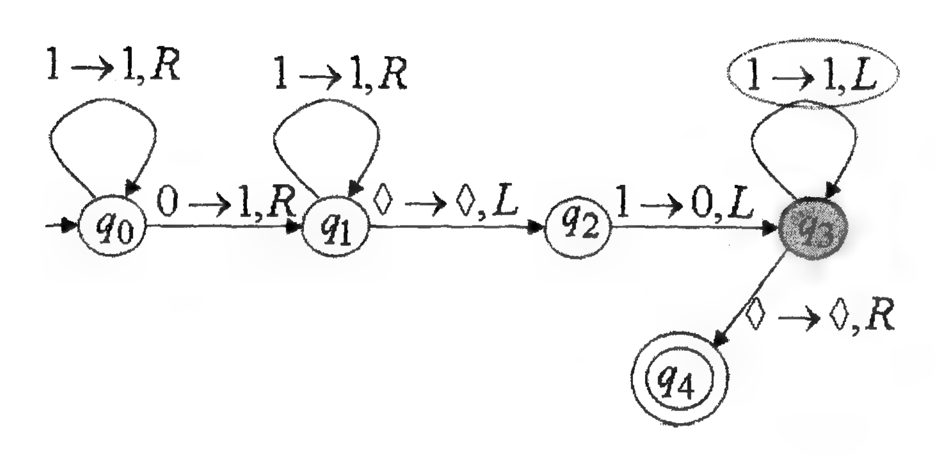

1→1,R
1→1,R
1→1,L
q_0
0→1,R
◊→◊,L
q_2
1→0,L
q_3
◊→◊,R
q_4

.7

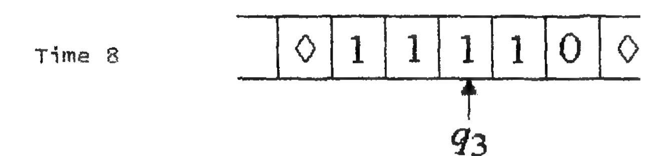

Time 6
◊
1
1
1
1
1
◊
q_2
1→1,R
1→1,R
1→1,L
q_0
0→1,R
q_1
◊→◊,L
1→0,L
q_3
◊→◊,R
q_4

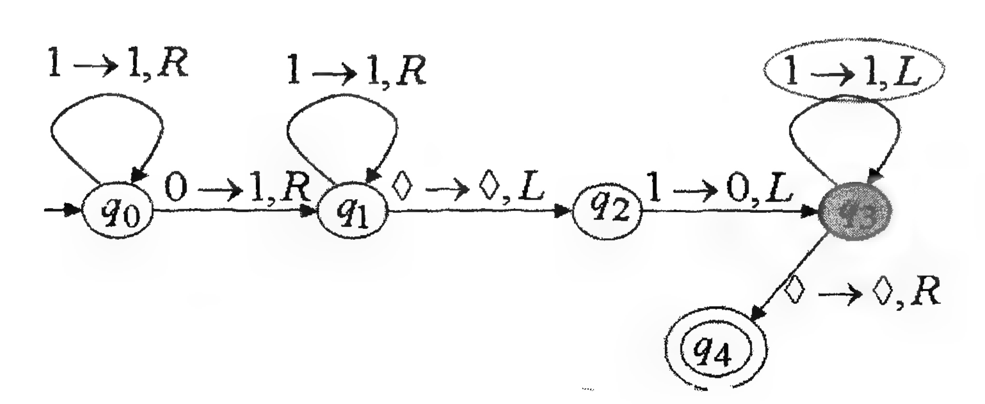
.8


Time 7
◊ 1 1 1 1 0 ◊
q_3

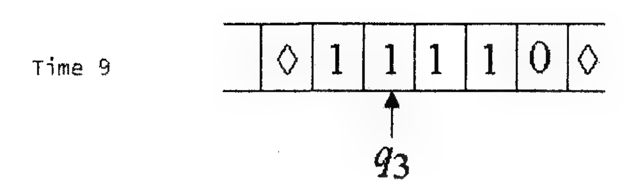

1 → 1, R
1 → 1, R
1 → 1, L
0 → 1, R
◊ → ◊, L
1 → 0, L
◊ → ◊, R
q_0
q_1
q_2
q_3
q_4

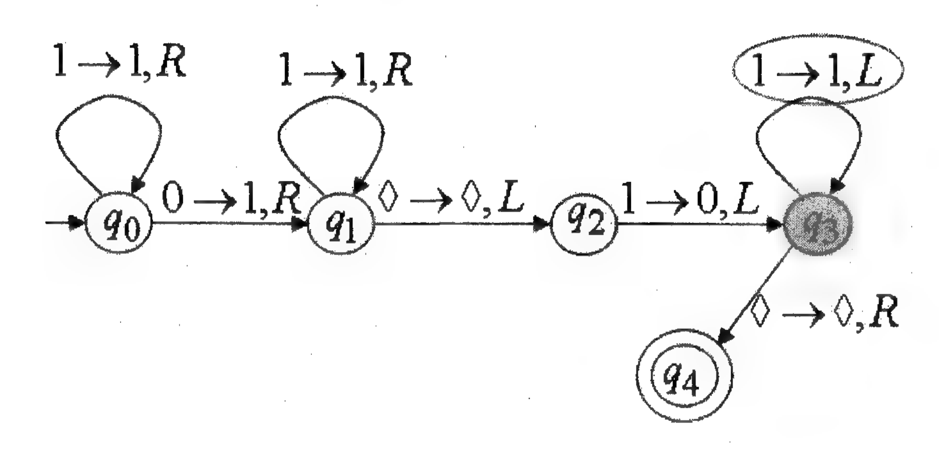
.9


Time 8
◊ 1 1 1 1 0 ◊
q_3

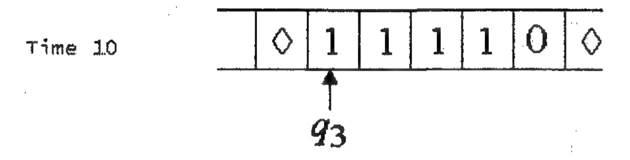

1 → 1, R
1 → 1, R
1 → 1, L
0 → 1, R
◊ → ◊, L
1 → 0, L
◊ → ◊, R
q_0
q_1
q_2
q_3
q_4

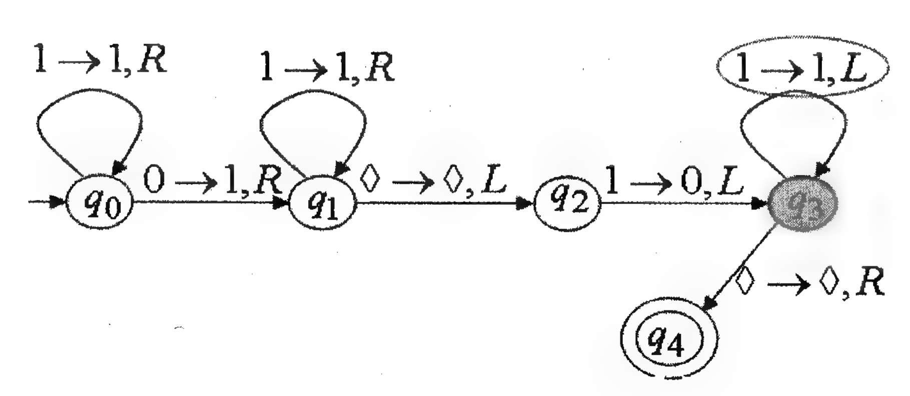
.10


Time 9
◊
1
1
1
1
0
◊
q_3

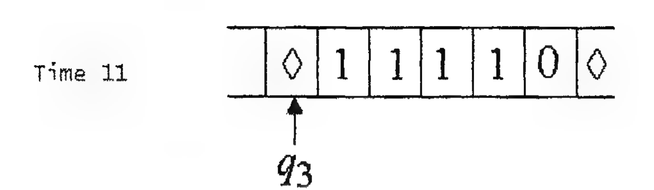

$1 \to 1, R$
$1 \to 1, R$
$1 \to 1, L$
q_0
$0 \to 1, R$
q_1
$◊ \to ◊, L$
q_2
$1 \to 0, L$
q_3
$◊ \to ◊, R$
q_4

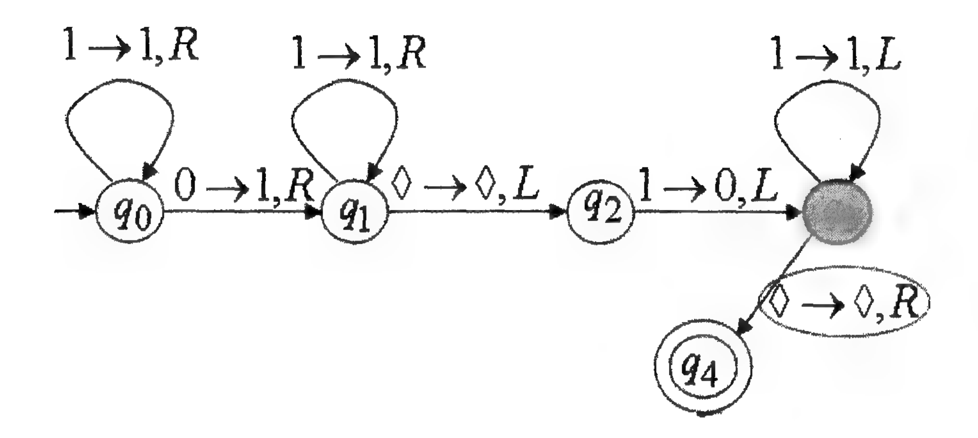
.11


Time 10
◊
1
1
1
1
0
◊
q_3

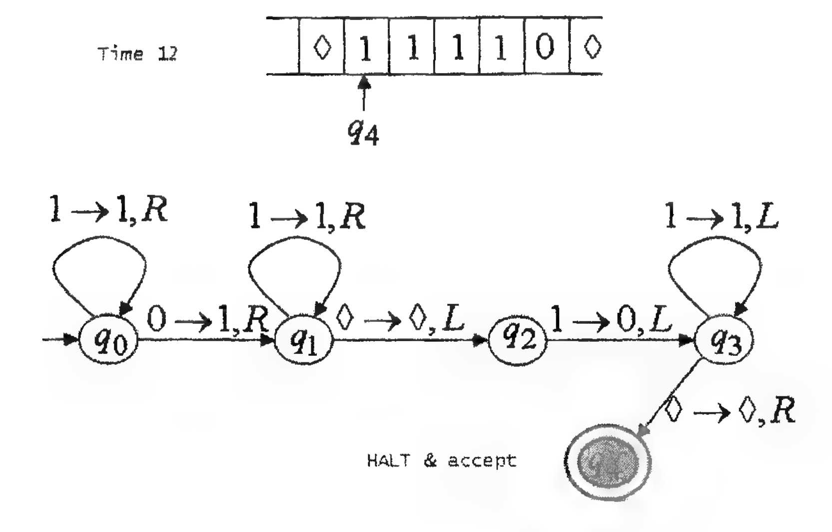

$1 \to 1, R$
$1 \to 1, R$
$1 \to 1, L$
q_0
$0 \to 1, R$
q_1
$◊ \to ◊, L$
q_2
$1 \to 0, L$
q_3
$◊ \to ◊, R$
q_4

.12


Time 11
◊
1
1
1
1
0
◊
q_3


1 → 1, R
1 → 1, R
1 → 1, L
0 → 1, R
◊ → ◊, L
1 → 0, L
◊ → ◊, R
q_0
q_1
q_2
q_4

.13


Time 12
◊
1
1
1
1
0
◊
q_4
1 → 1, R
1 → 1, R
1 → 1, L
0 → 1, R
◊ → ◊, L
1 → 0, L
◊ → ◊, R
q_0
q_1
q_2
q_3
HALT & accept

مثال:

The function $f(x) = 2x$ is computable

x is integer

Turing Machine:

Input string: x unary

Output string: xx unary

x

Start	◊	1	1	Λ	1	◊

q_0 initial state

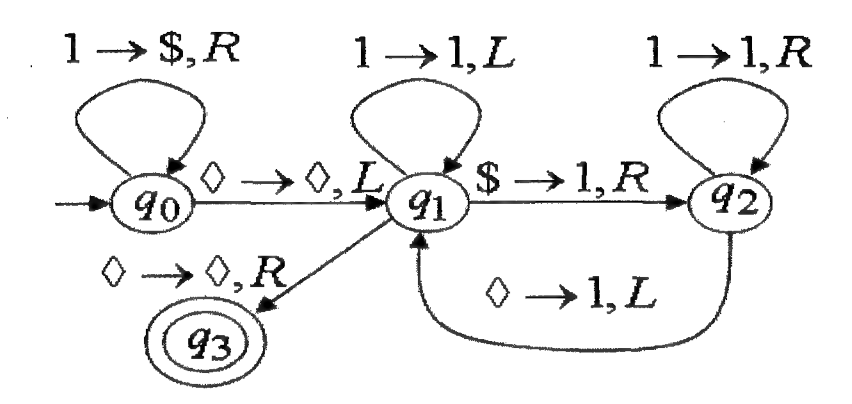
$2x$

Finish	◊	1	1	Λ	1	1	1	◊

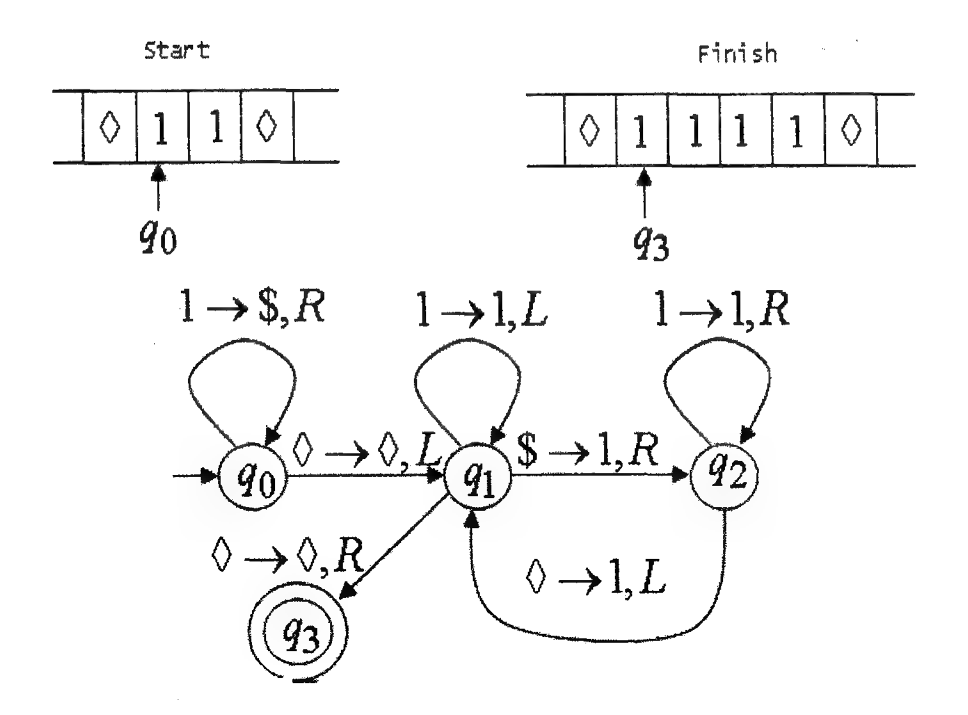
q_f final state

وفيما يلي الة تيورينج للتعرف على هذا الإقتران:

Turing Machine for $f(x) = 2x$



وفيما يلي الية تنفيذ هذه الالة:



سؤال:

ابن الة تيورينج للتعرف على الاقتران التالي:

The function

$$f(x,y)=\begin{cases}1 & \text{if} \quad x>y \\ 0 & \text{if} \quad x\le y\end{cases}$$

is computable

Turing Machine for $f(x,y)=\begin{cases}1 & \text{if} \quad x>y \\ 0 & \text{if} \quad x\le y\end{cases}$

Input: $x0y$

Output: 1 or 0

تستخدم الة تيورينج ايضا في تطبيقات علم الحاسوب وخاصة في تمثيل العمليات المنفذة من قبل معالجات النصوص ومن هذه العمليات:

- نسخ الرموز أو السلسلة الرمزية.
- نقل و تحريك النصوص.

- استبدال الرموز برموز اخرى.
- إزاحة الرموز لليسار.
- إزاحة الرمموز لليمين.
- نفل رأس القراءة والكتابة الى رمز معين أو عملية البحث عن أول حرف أورمز.
- تكرار اي من العمليات السابقة عدد محدد من المرات.

وفيما يلي نستعرض بعض الامثلة والتي تبين كيفية استخدام الة تيورينج لتمثيل بعض العمليات المنفذة من قبل برمجيات معالجات النصوص أو ما يسمى برامج تحرير النص.

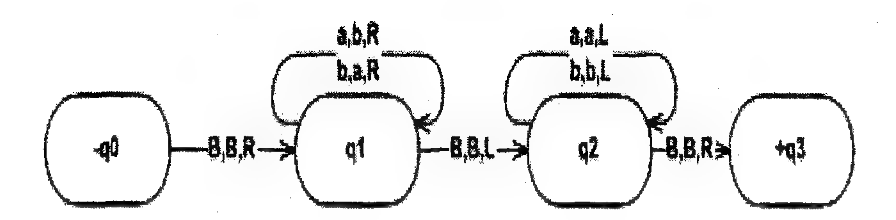
مثال:

صمم الة تيورينج والتي تعالج مجموعة من الرموز مؤلفة من حرفين بيحث تستبدل الحرف الاول بالثاني والحرف الثاني بالاول.

نستعرض الية بناء هذه الالة من خلال تسلسل التنفيذ فيها والذي سيكون كما يلي:

BbabB (character about to be read)
BbabB
BaabB
BabbB
BabaB
BabaB
BabaB
BabaB
BabaB
BabaB (halts pointing at 1st output character)

وفيما الة تيورينج لتنفيذ هذه العملية:



مثال:

عملية النسخ

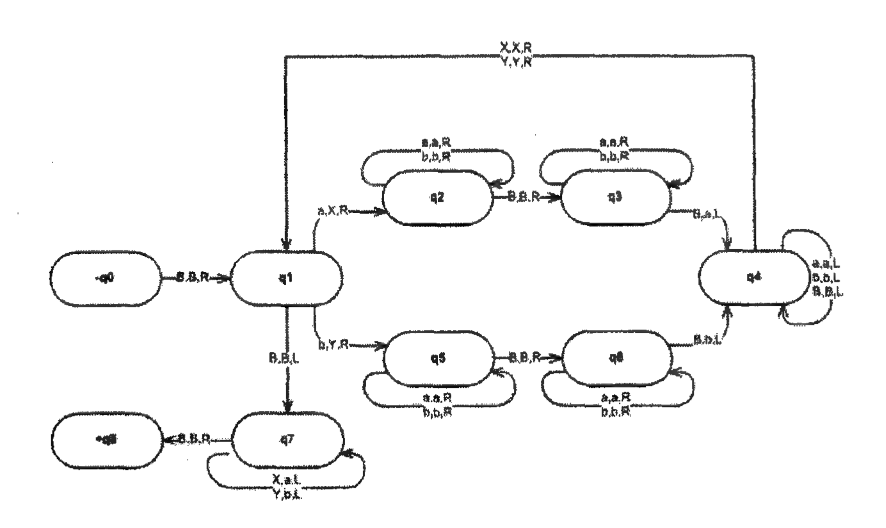
تتلخص عملية نسخ مجموعة من الرموز في الخطوات التالية:

1. نقل رأس القراءة والكتابة الى أول فراغ في اليسار.
2. التحرك لليمين لخطوة واحدة.
3. كتابة فراغ أو أي رمز بدل الحرف والاحتفاظ به.
4. البحث عن ثاني فراغ باتجاه اليمين.
5. كتابة الرمز في الموقع.
6. البحث عن ثاني فراغ باتجاه اليسار .
7. اعادة كتابة الرمز بدل الفراغ.
8. تكرار الخطوات 2 الى 7 ولكل رمز أو حرف.

فلو أخذنا حرفين وكما هو مبين ادناه فان الية التنفيذ ستتم كما يلي:

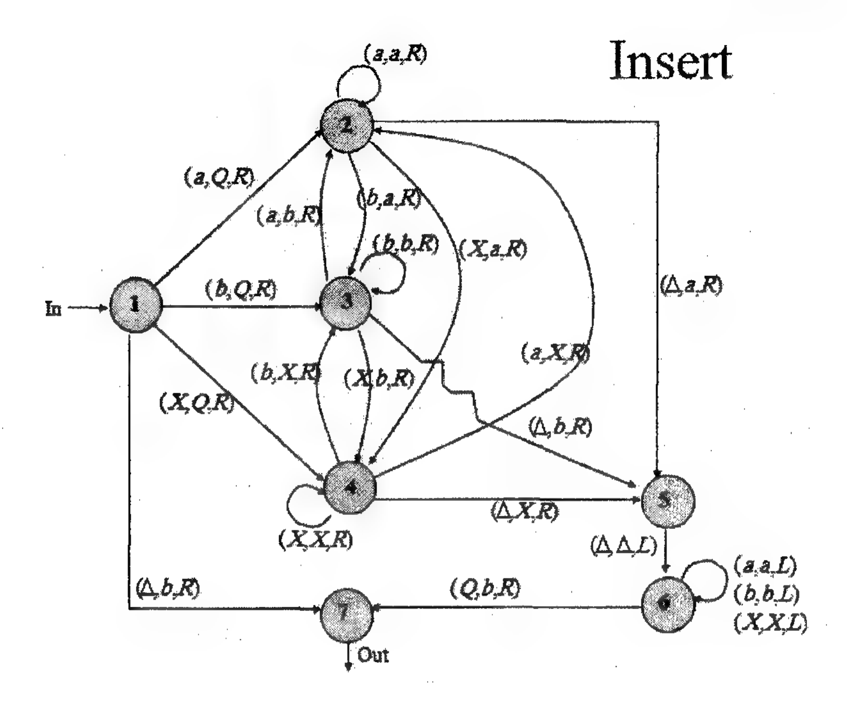
BabB
BabB
BXbB
BXbB
BXbBB
BXbBa
BXbBa
BXbBa
BXbBa
BXYBa
BXYBa
BXYBaB
BXYBab
BXYBab
BXYBab
BXYBab
BXYBab
BXbBab
BabBab (halt)

اما الة تيورينج والمنفذة لهذه العمليات فهي كما يلي:



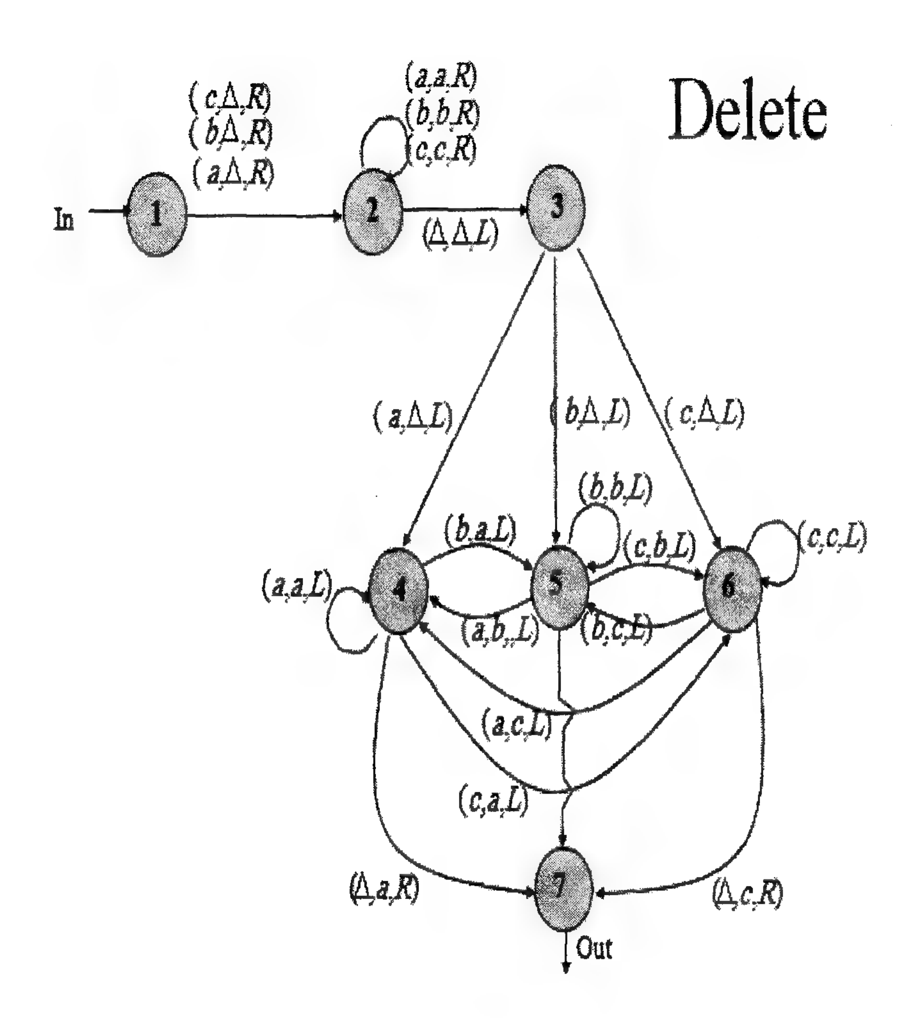
مثال:

الة الادخال



مثال:

الة الحذف



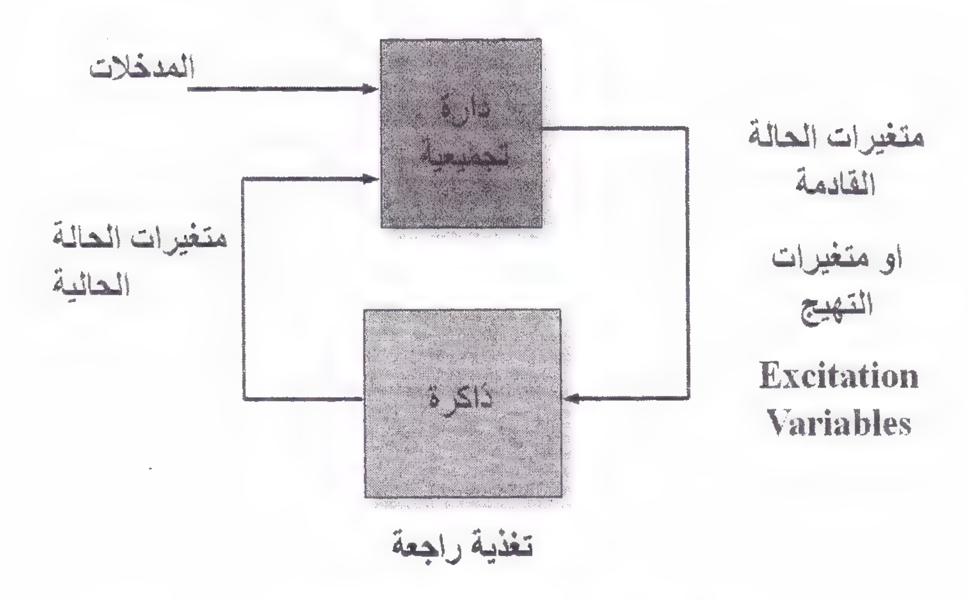


الوحدة السادسة
آلة مورو آلة ميلي
Moore &Mealy
Machine
6

1.6 مقدمة:

تستخدم الة الحالة المنتهية كما اشرنا سابقا في هذا الكتاب في تطبيقات متعددة ومن هذه التطبيقات تصميم الدارات المنطقية التتابعية والمتزامنة.

تمثل الدارة المنطقية التتابعية المتزامنة الة حالة منتهية وتتكون هذه الالة في الغالب من دارة منطقية تجميعية مؤلفة من البوابات المنطقية المختلفة ومن ذاكرة للاحتفاض بحالة الدراة الحالية ويبين الشكل التالي معمار الة الحالة المنتهية والتي يمكن استخدامها في عملية التصميم المنطقي لدارات المنطق التتابيعة:



يتم بناء الذاكرة في الة الحالة المنتهية باستخدام أحد انواع النطاطات Flip-Flops ويتوفر من هذه النطاطات اربعة أنواع هي:

- النطاط RS
- النطاط JK
- النطاط D
- النطاط T

ولإستخدام أي من هذه النطاطات في تصميم الة الحالة المنتهية لا بد من معرفة بعض الامور الاساسية عن كل نطاط ونخص بالذكر:

- مجموعة المداخل والمخارج.
- جدول الصواب والذي يبين كيفية انتقال النطاط من الحالة الحالية الى الحالة التالية بوجود نبضة الساعة وبعض القيم على المداخل.
- المعادلة المميزة والتي تبين الحالة التالية كإقتران يعتمد على قيم المداخل وقيمة الحالة الحالية.
- جدول التهيج والذي يستخدم في عملية تصميم الالة المنتهية والذي يبين القيم الواجب توفرها على المداخل لنقل النطاط من الحالة الحالية الى الحالة التالية.

يمتلك أي نطاط مجموعة من المداخل وكل نطاط يمكن ان يقع في اللحظة الزمنية اما في حالة الصفر أو في حالة الواحد ويمكن لهذا النطاط ان يغير حالته التالية أو ان يبقى ثابتا عليها اعتمادا على قيم المداخل وقيمة الحالة الحالية.

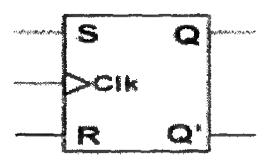
من المداخل المستخمة في كافة انواع النطاطات مدخل المسح Clear ومدخل النقل الى الواحد Preset

فوجود 1 على المدخل الاول و صفر على المدخل الثاني يؤدي الى نقل النطاط الى الحالة صفر بغض النظر عن القيم الاخرى على المداخل اما وجود الصفر على المدخل الاول و واحد على المدخل الثاني فان هذا يؤدي الى نقال النطاط الى الحالة 1 بغض النظر عن القيم الموجودة على المداخل الاخرى، ولإستخدام النطاط في عملية التصميم وخاصة في تصميم الة الحالة المنتهية تستخدم الحالة عندما تكون قيمة المدخل الاول مساوية لقيمة المدخل الثاني ومساوية للصفر و في

هذه الحالة فان الحالة التالية تتاثر بقيمة الحالة الحالية وقيم المداخل الاخرى بوجود نبضة الساعة.

النطاط SR

يبين الشكل التالي المخطط الصندوقي لهذا النطاط:



ينتقل هذا النطاط من الحالة الحالية الى الحالة المقبلة أو التالية بوجود قيم المداخل التالية والمبينة في جدول الصواب التالي لهذا النطاط:

S	*R*	*Q(next)*
0	0	Q
0	1	0
1	0	1
1	1	?

المعادلة المميزة والتي تربط الحالة التالية بقيم المداخل والحالة الحالية هي:

$$Q\ (next) = S + R'Q$$

$$SR = 0$$

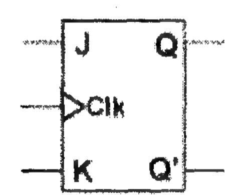
جدول التهيج والذي يحدد القيم الواجب توفرها على المداخل لنقل النطاط من الحالة الحالية الى حالة تالية معروفة هو:

Q	Q(next)	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

النطاط JK

يشبه هذا النطاط النطاط السابق الا انه يقبل القيم 1 علة الماخل وفي هذه الحالة يعمل النطاط على عكس الحالة الحالية في اللحظة الزمنية التالية.

يبين الشكل التالي المخطط الصندوقي لهذا النطاط



ينتقل هذا النطاط من الحالة الحالية الى الحالة المقبلة أوالتالية بوجود قيم المداخل التالية والمبينة في جدول الصواب التالي لهذا النطاط:

J	K	Q(next)
0	0	Q
0	1	0
1	0	1
1	1	Q'

المعادلة المميزة والتي تربط الحالة التالية بقيم المداخل والحالة الحالية هي:

$$Q\,(next) = JQ' + K'Q$$

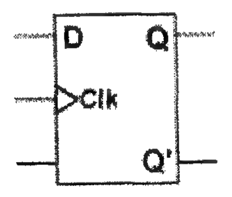
جدول التهيج والذي يحدد القيم الواجب توفرها على المداخل لنقل النطاط من الحالة الحالية الى حالة تالية معروفة هو:

Q	*Q(next)*	*J*	*K*
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

النطاط D

نطاط يشبه الى حد ما النطاط JK لكن بوصل المدخل J مع K عن طريق بوابة نفي.

يبين الشكل التالي المخطط الصندوقي لهذا النطاط:



ينتقل هذا النطاط من الحالة الحالية الى الحالة المقبلة أوالتالية بوجود قيم المداخل التالية والمبينة في جدول الصواب التالي لهذا النطاط:

D	*Q(next)*
0	0
1	1
0	0

المعادلة المميزة والتي تربط الحالة التالية بقيم المداخل والحالة الحالية هي:

Q (next) = D

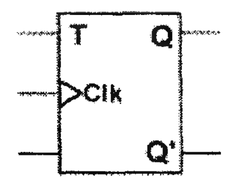
جدول التهيج والذي يحدد القيم الواجب توفرها على المداخل لنقل النطاط من الحالة الحالية الى حالة تالية معروفة هو:

Q	*Q(next)*	*D*
0	0	0
0	1	1
1	0	0
1	1	1

النطاط T

نطاط يشبه الى حد ما النطاطJK لكن بوصل المدخل J مع K.

يبين الشكل التالي المخطط الصندوقي لهذا النطاط:



ينتقل هذا النطاط من الحالة الحالية الى الحالة المقبلة أوالتالية بوجود قيم المداخل التالية والمبينة في جدول الصواب التالي لهذا النطاط:

T	*Q(next)*
0	Q
1	Q'
T	Q(next)

المعادلة المميزة والتي تربط الحالة التالية بقيم المداخل والحالة الحالية هي:

$$Q\ (next) = TQ' + T'Q$$

جدول التهيج والذي يحدد القيم الواجب توفرها على المداخل لنقل النطاط من الحالة الحالية الى حالة تالية معروفة هو:

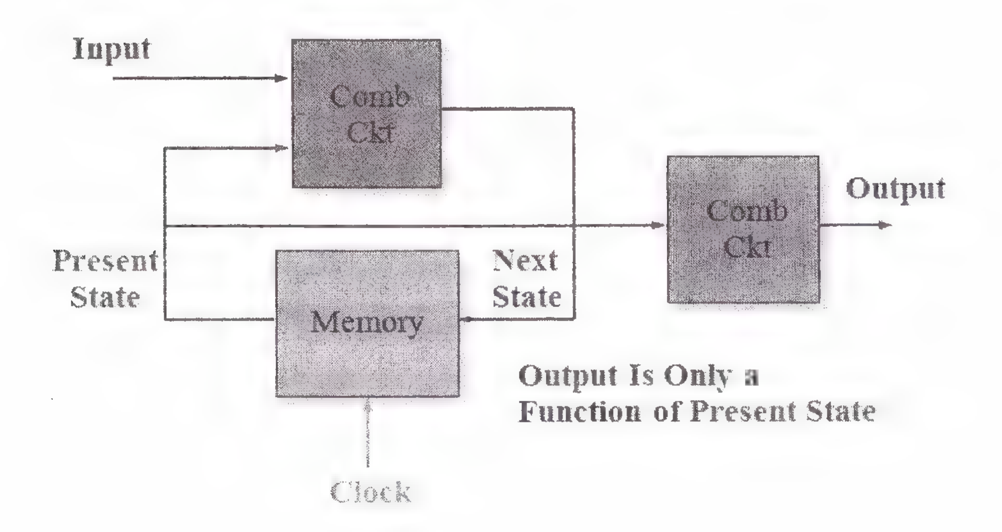
Q	*Q(next)*	*T*
0	0	0
0	1	1
1	0	1
1	1	0

وفيما يلي جدول يبين ملخص جداول التهيج للنطاطات السابقة والتي يمكن استخدامها في عملية التصميم:

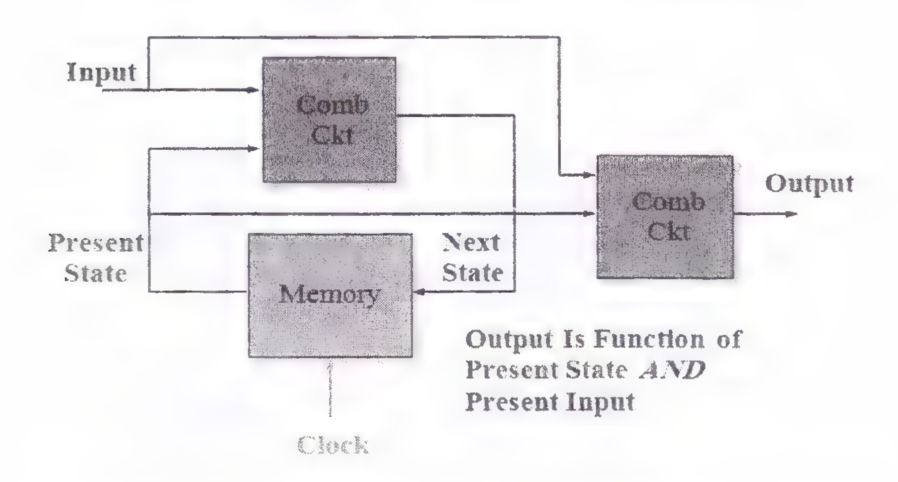
Desired transition		Triggering signal needed					
Q(t)	Q(t+1)	S	R	J	K	D	T
0	0	0	x	0	x	0	0
0	1	1	0	1	x	1	1
1	0	0	1	x	1	0	1
1	1	x	0	x	0	1	0

2.6 الة مورو الة ميلي:

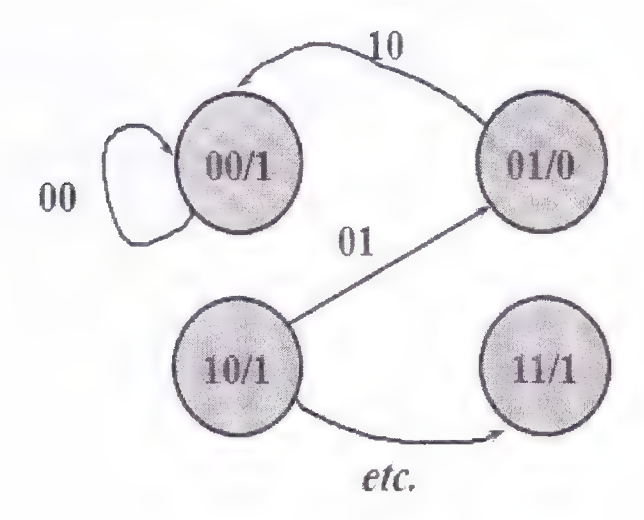
تعتبر هاتان الالتان نماذج من الة الحالة النتهية ويتالف كل منها من دارة تجميعية وذاكرة وتشبه الة مور الة ميلي لكن بخلاف بسيط في دالة الانتقال لكل منهما ففى الة مور يتم تواليد المخرجات عند استقرار الالة في حالة معينة اي ان المخرجات تشكل اقترانا يعتمد على الحالة الحالية اما في الة ميلي فيتم توليد المخرجات عند انتقال الالة من حالة الى اخرى اي ان المخرجات هي اقترانات تعتمد على قيم المدخلات الحالية وقيمة الحالة الحالية ويبين الشكلان التاليان الخلاف بين هاتين الالتين:



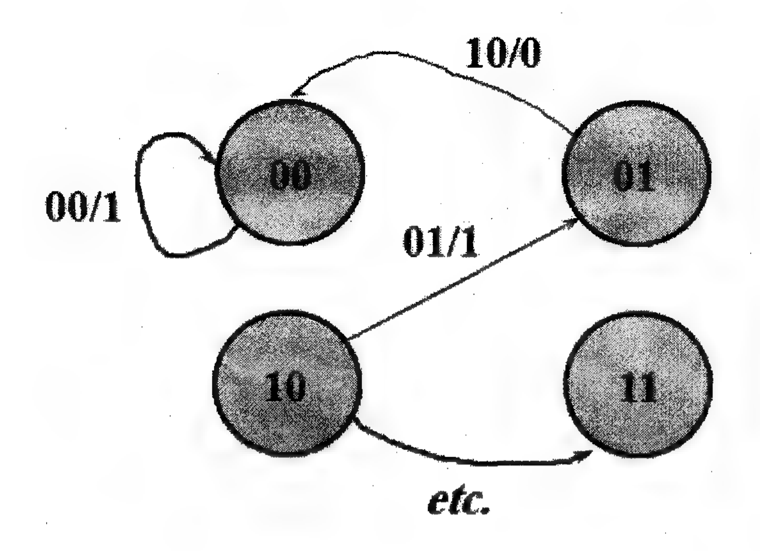
Synchronous Mealy Machine



تمثل الة مور بمخطط الحالات وذلك بوضع المخرجات في الحالة وكما يلي:



اما في الة ميلي فيتم تمثيل المخرجات عند الإنتقال من حالة الى اخرى كما يلي:



تمثل كل من الة مورو الة ميلي بنموذج رياضي يضم:

- مجموعة منتهية من الحالات.
- مجموعة منتهية من المدخلات.
- مجموعة منتهية من المخرجات.
- دالة الانتقال من حالة لاخرى.
- دالة المخرجات.

وفيما يلي النموذج الرياضي لكل من الة مورو الة ميلي:

Machine is a quintuple of sets

$M = (S, I, O, \delta, \beta)$

S: Finite set of states

I: Finite set of inputs

O: Finite set of outputs

δ: State transition function

β/λ: Mealy/Moore output function

ولتحديد عناصر هذا النموذج ولأي من الالتين لا بد من اتباع الخطوات التالية:

- فهم المشكلة وتحديد مدخلاتها ومخرجاتها.
- تمثيل المشكلة باستخدام مخطط الحالات.
- تحديد جدول الإنتقال.

مثال:

حدد النموذج الرياضي لكل من الة ميلي والة مورواللازمة لاكتشاف التتابع 0101 في مجموعة من المدخلات المؤلفة من الصفر والواحد.

الة ميلي:

- مكونات النموذج.

$$M = (S, I, O, \delta, \beta)$$

S: { A, B, C, D }

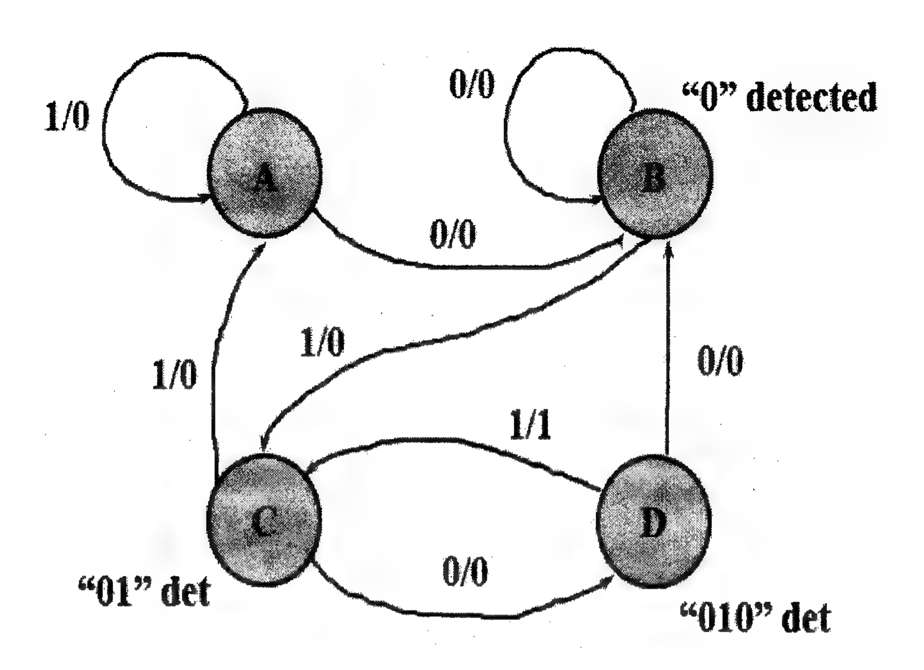
I: { 0, 1 }

O: { 0, 1 } = { not detected, detected}

δ: next figure

β: next figure

- مخطط الحالات:



- جدول الانتقال(دوال الانتقال):

Present State	Present Input	
	0	1
A	B/0	A/0
B	B/0	C/0
C	D/0	A/0
D	B/0	C/1

Next State/Output

آلة مور:

- مكونات النموذج:

$$M = (S, I, O, \delta, \lambda)$$

S: { A, B, C, D, E }

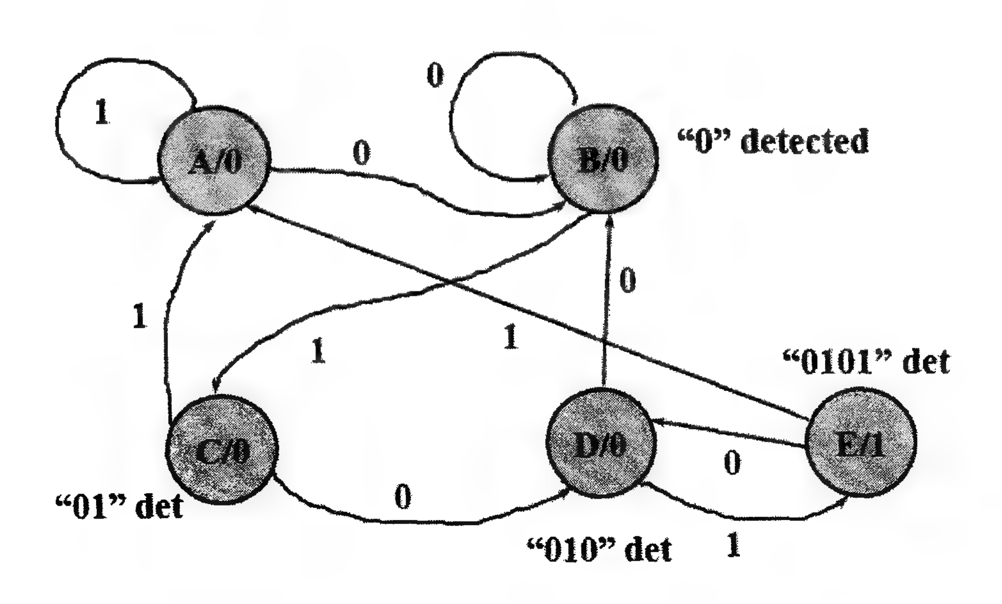
I: { 0, 1 }

O: { 0, 1 } = { not detected, detected}

δ: next figure

λ: next figure

- مخطط الحالات:



- جدول الانتقال:

Present State	Present Input		Output(λ)
	0	1	
A	B	A	0
B	B	C	0
C	D	A	0
D	B	E	0
E	D	A	1

Next State

3.6 تصميم الة الحالة المنتهية:

تستخدم في عملية تصميم أي من الة ميلي أو مور الخطوات الاساسية التالية:

1. فهم المشكلة بتحديد المدخلات والمخرجات وكيفية توليدها.
2. تمثيل المشكلة بمخطط الحالات.
3. استخدام النظام الثنائي في ترقيم الحالات لتحديد عدد النطاطات المطلوبة.
4. تحديد نوع النطاط المراد استخدامه في عملية التصميم.
5. بناء جدول الانتقال وباستخدام جدول التهيج للنطاط.
6. استخراج المعادلات المختصرة لكل مدخل من مداخل النطاط واستخراج المعادلة المختصرة للمخرجات.
7. تمثيل الالة باستخدام بوابات ودارات المنطق.

مثال:

صمم الة الحالة المنتهية لاكتشاف 3 وحدات متتابعة في مجموعة من المدخلات المؤلفة من الصفر والواحد.

الحل:

- نحدد الحالات لالة مور كما يلي:

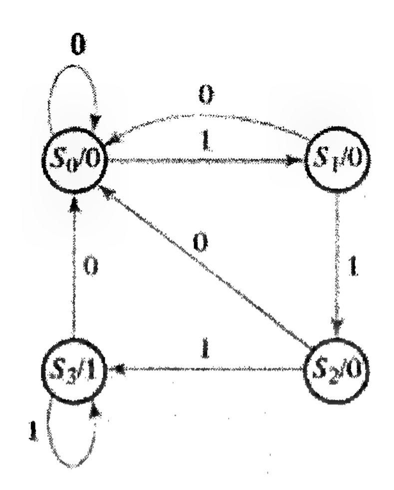
State S_0 : zero 1s detected

State S_1 : one 1 detected

State S_2 : two 1s detected

State S_3 : three 1s detected

- نرسم مخطط الحالات:



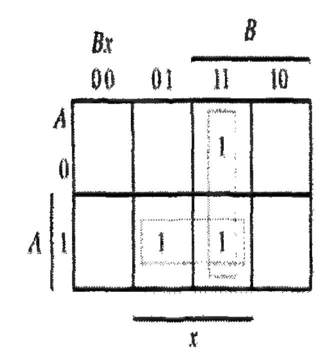
- نرقم الحالات باستخدام النظام الثنائي:

- $S_0 = 00$
- $S_2 = 10$
- $S_1 = 01$
- $S_3 = 11$

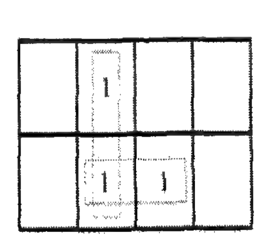
- عدد النطاطات المطلوب هو 2 ولنخترالنطاط D.
- نبني جدول الانتقال لاحظ هنا ان مدخل النطاط يساوي الحالة المقبلة:

Present State		Input	Next State		Output
A	B	x	A	B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

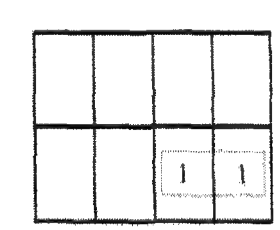
- نستخدم خرائط كارنو للحصول على المعادلات المختصرة لمداخل النطاطات ودالة المخرج:



$D_A = Ax + Bx$

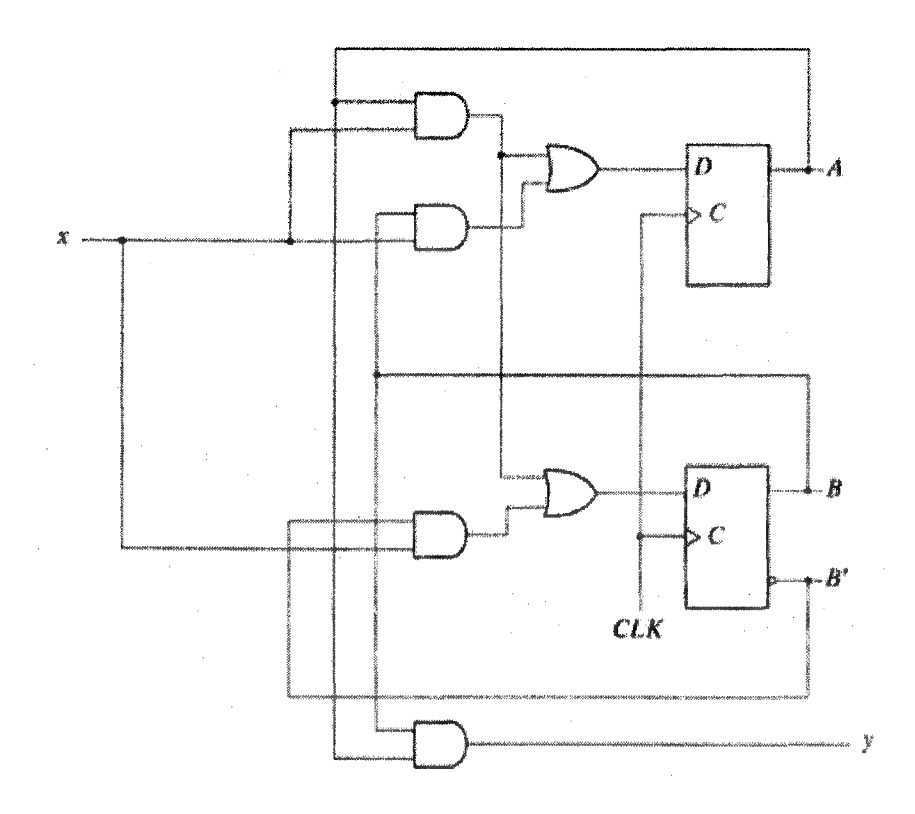


$D_B = Ax + B'x$



$y = AB$

- بإستخدام المعادلات المختصرة نبني آلة مور المنطقية باستخدام دارات المنطق:

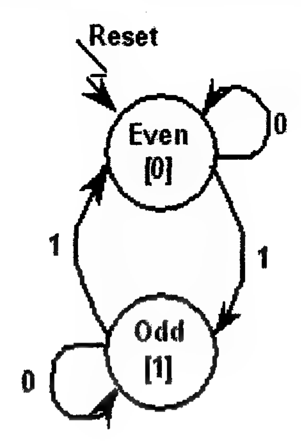


مثال:

فحص التطابق الفردي

فيما يلي الة مور لفحص التطابق الفردي وانتاج 1 عندما يكون عدد الوحدات المقروءة فرديا:

- مخطط الحالات



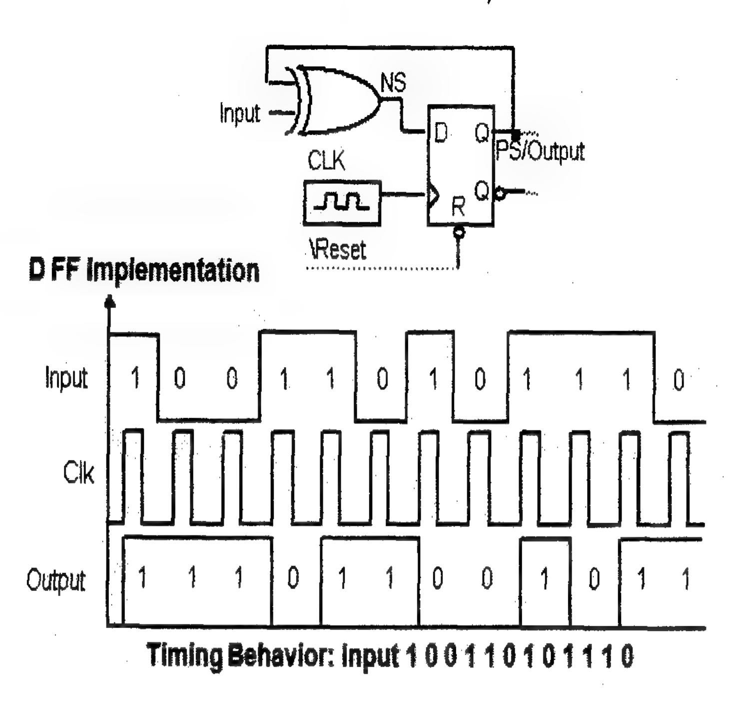
- نبني جدول الانتقال.

Present State	Input	Next State	Output
Even	0	Even	0
Even	1	Odd	0
Odd	0	Odd	1
Odd	1	Even	1

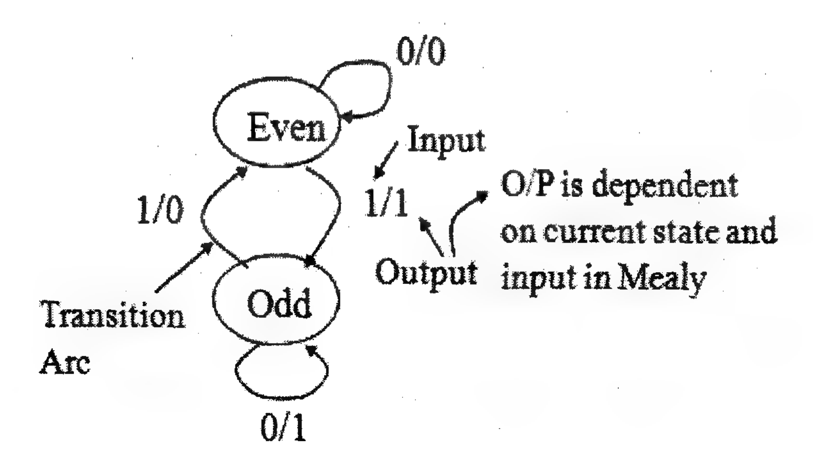
Symbolic State Transition Table

Present State	Input	Next State	Output
0	0	0	0
0	1	1	0
1	0	1	1
1	1	0	1

- نحدد المعادلات ونبني الدارة المنطقية.



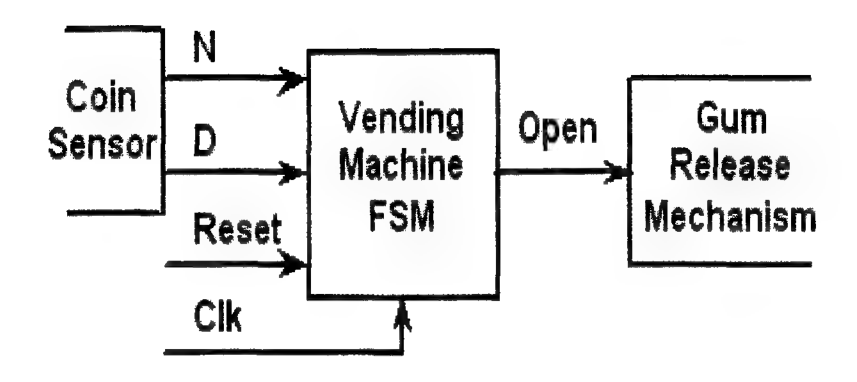
اما مخطط الحالات لآلة ميلي فهو كما يلي:



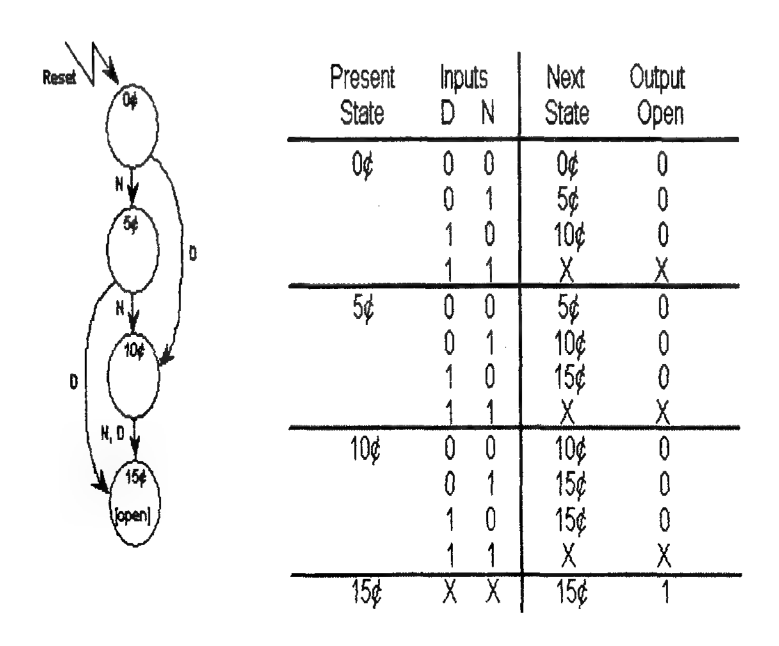
مثال:

تصميم وحدة ذاتية لإعطاء القهوة في الة القهوة الاوتوماتيكية.

يمكن تصور هذه الالة كما يلي:



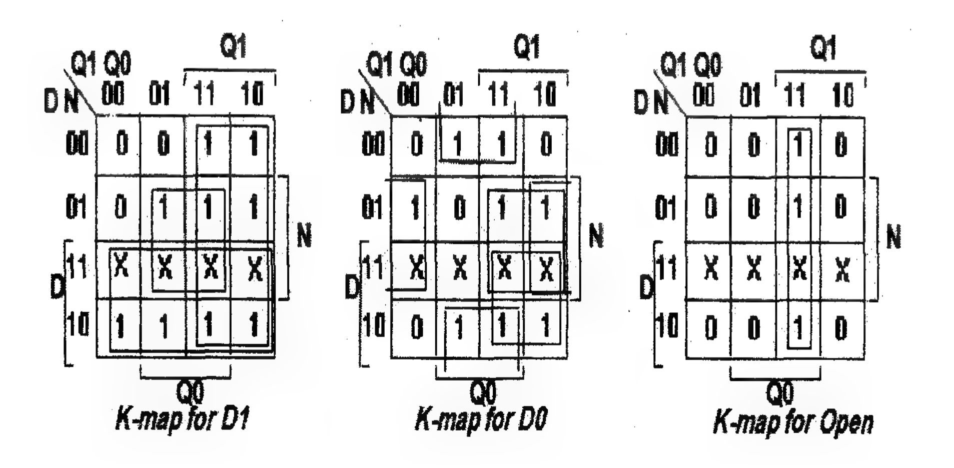
- مخطط الحالات وجدول الانتقال:



Present State	Inputs D	Inputs N	Next State	Output Open
0¢	0	0	0¢	0
	0	1	5¢	0
	1	0	10¢	0
	1	1	X	X
5¢	0	0	5¢	0
	0	1	10¢	0
	1	0	15¢	0
	1	1	X	X
10¢	0	0	10¢	0
	0	1	15¢	0
	1	0	15¢	0
	1	1	X	X
15¢	X	X	15¢	1

Present State Q_1	Q_0	Inputs D	N	Next State D_1	D_0	Output Open
0	0	0	0	0	0	0
		0	1	0	1	0
		1	0	1	0	0
		1	1	X	X	X
0	1	0	0	0	1	0
		0	1	1	0	0
		1	0	1	1	0
		1	1	X	X	X
1	0	0	0	1	0	0
		0	1	1	1	0
		1	0	1	1	0
		1	1	X	X	X
1	1	0	0	1	1	1
		0	1	1	1	1
		1	0	1	1	1
		1	1	X	X	X

- اختصار المعادلات:

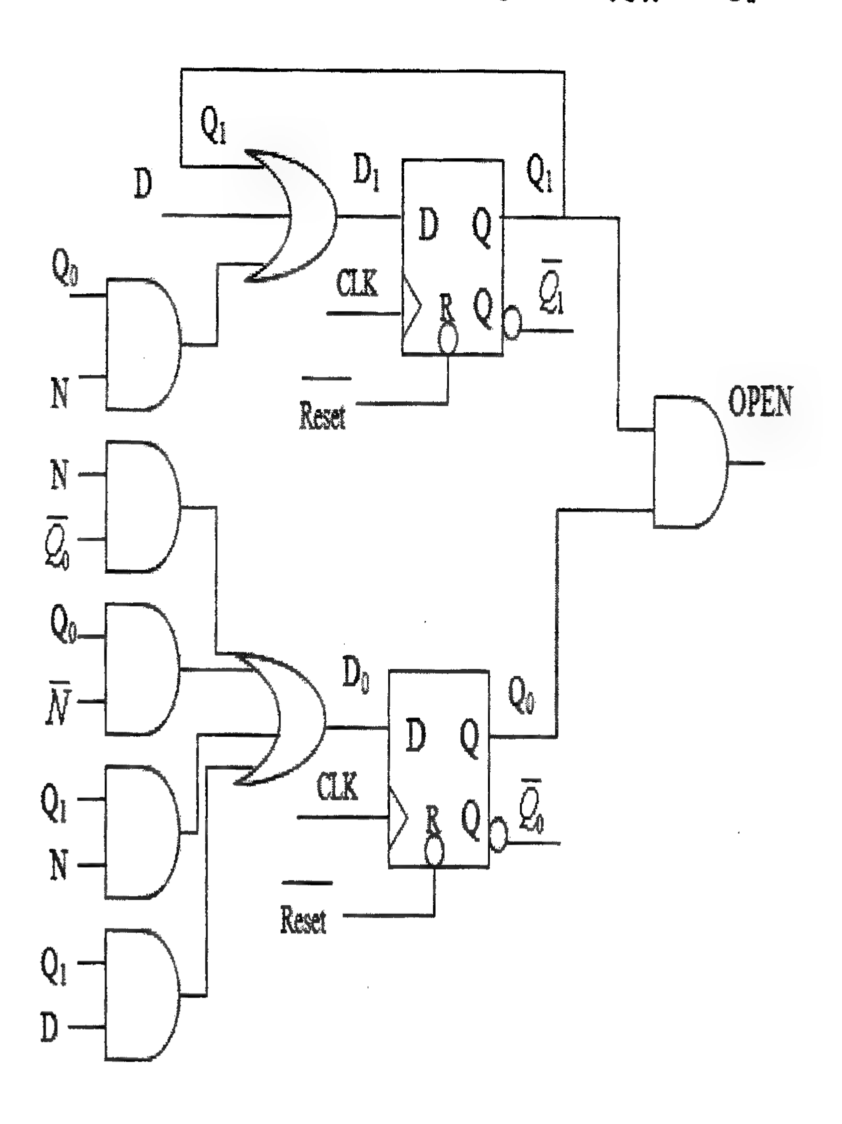


K-map for D1

K-map for D0

K-map for Open

- تمثيل الالة ببوابات المنطق:



التصميم باستخدام النطاط JK

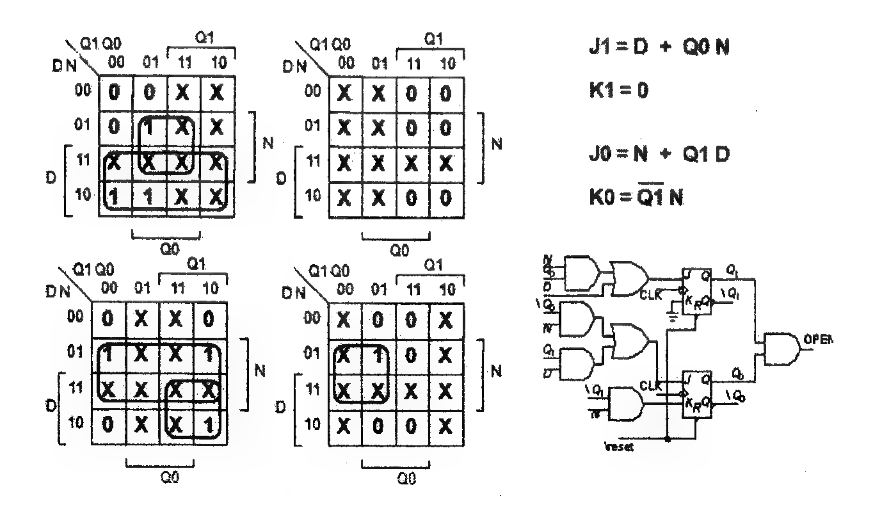
هذه الحالة نستخدم جدول التهيج لهذا النطاط لبناء جدول الانتقال:

JK Excitation Table

Q	Q+	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Present State Q_1	Q_0	Inputs D	N	Next State Q_1^+	Q_0^+	J_1	K_1	J_0	K_0
0	0	0	0	0	0	0	X	0	X
		0	1	0	1	0	X	1	X
		1	0	1	0	1	X	0	X
		1	1	X	X	X	X	X	X
0	1	0	0	0	1	0	X	X	0
		0	1	1	0	1	X	X	1
		1	0	1	1	1	X	X	0
		1	1	X	X	X	X	X	X
1	0	0	0	1	0	X	0	0	X
		0	1	1	1	X	0	1	X
		1	0	1	1	X	0	1	X
		1	1	X	X	X	X	X	X
1	1	0	0	1	1	X	0	X	0
		0	1	1	1	X	0	X	0
		1	0	1	1	X	0	X	0
		1	1	X	X	X	X	X	X

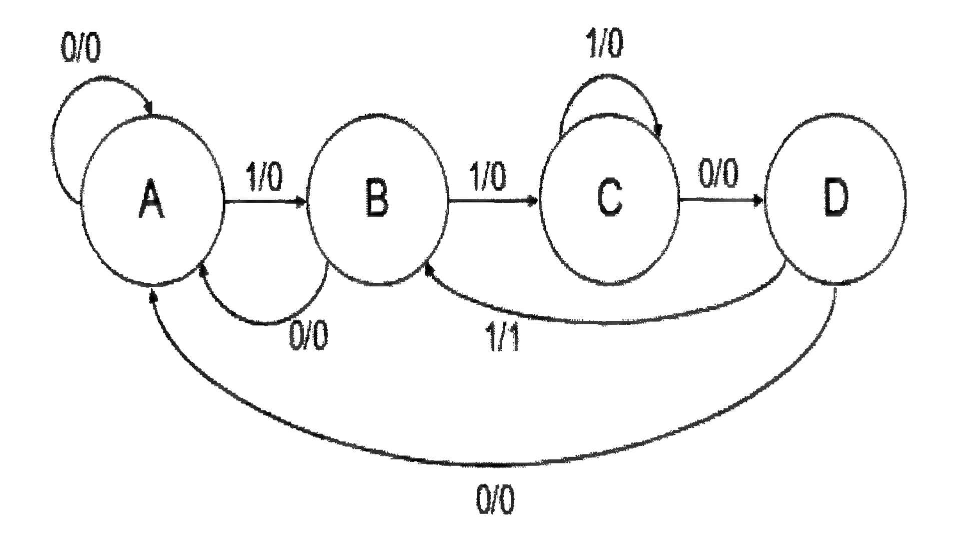
نستخرج المعادلات ز نرسم المخطط المنطقي للالة:



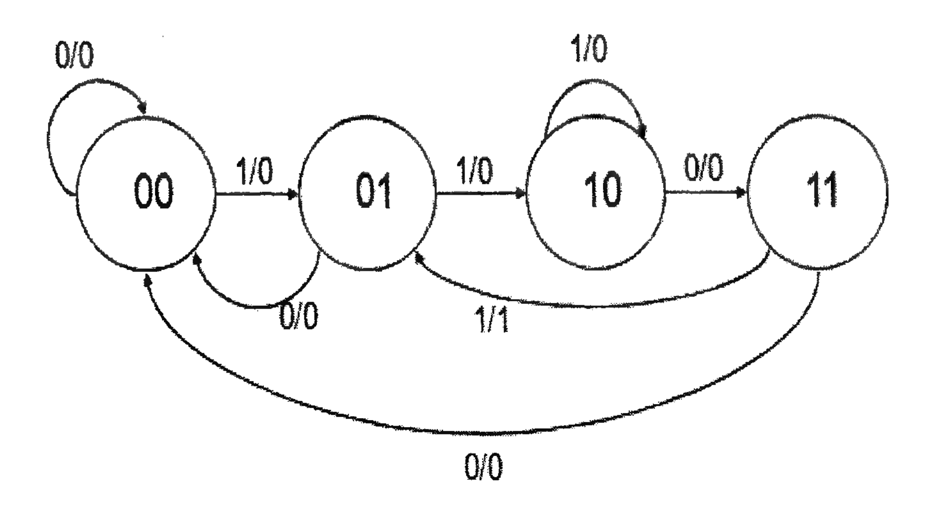
مثال:

صمم الة ميلي لتوليد 1 بعد قراءة 1101 في مجموعة من المدخلات المؤلفة من الصفر والواحد(او لإكتشاف النمط 1101)

- مخطط الحالات:



- نرقم الحالات:



- نبني جدول الانتقال:

Present State		Input	Next State		Output
A	B	X	A	B	Y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	1	1	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	1	0	1	1

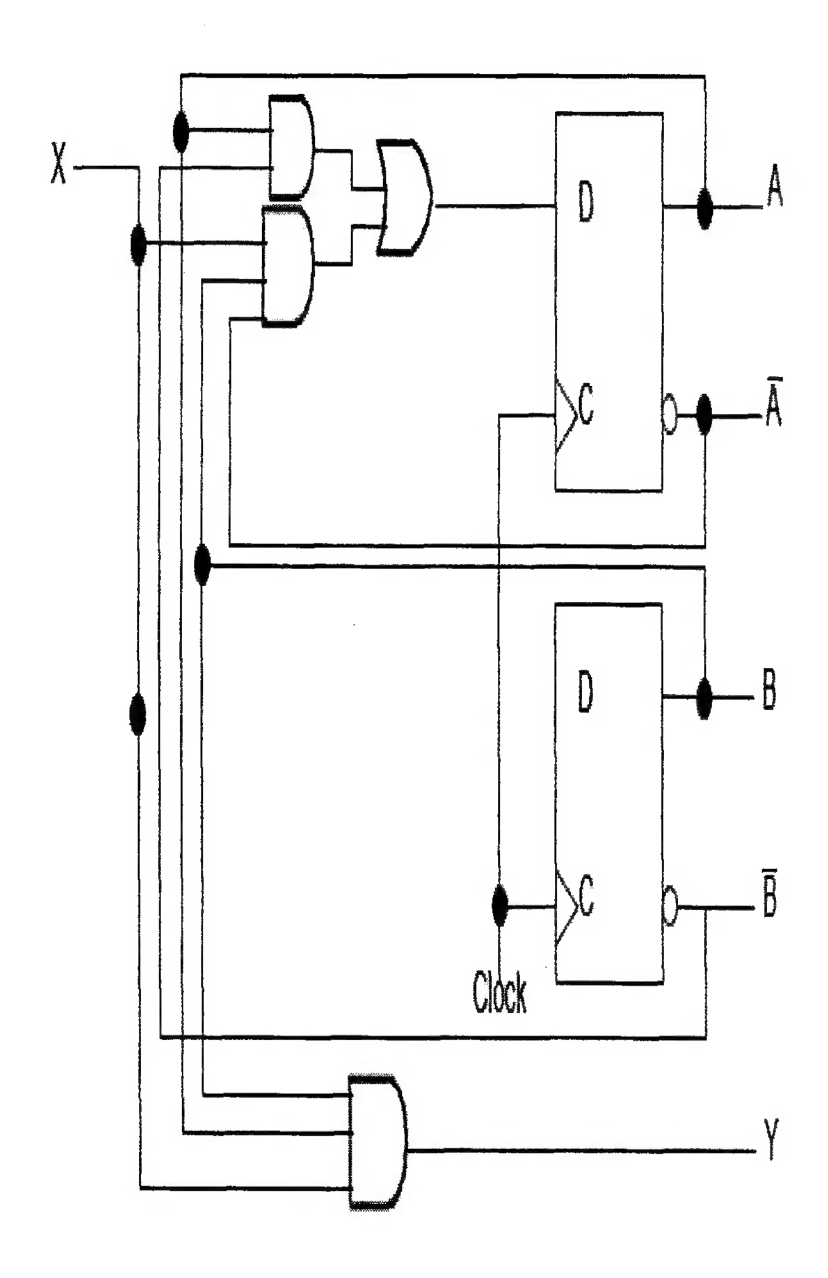
- نستخرج المعادلات:

$$A_{next} = A'BX + AB'$$

$$B_{next} = A'B'X + AB'X' + ABX$$

$$Y = ABX$$

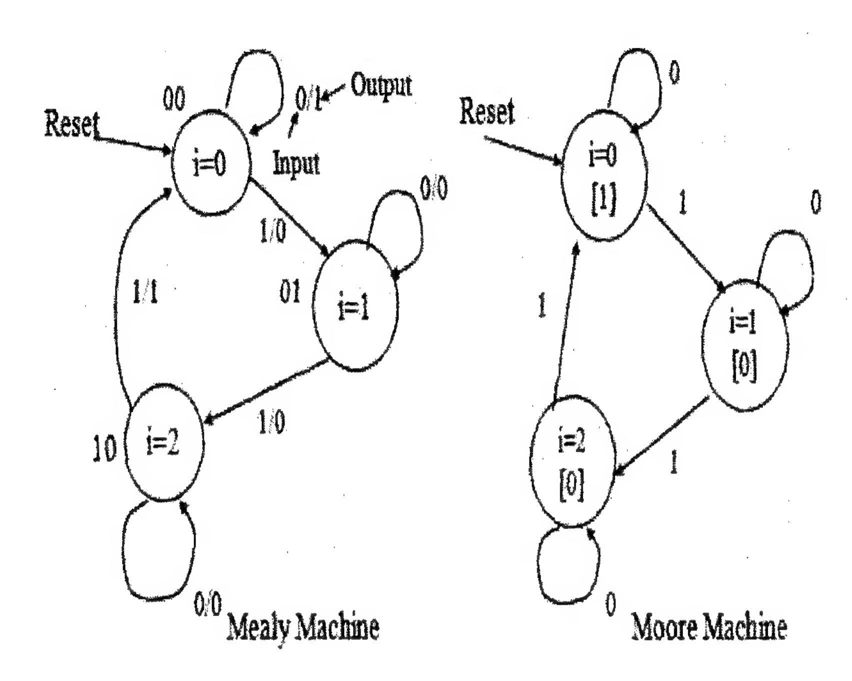
- نمثل الالة ببوابات المنطق:



سؤال:

صمم الة الحالة لتوليد 1 عند يكون عدد الوحدات المقروءة من مضاعفات الرقم 3.

استعن بمخطط الحالات التالي:



References

Barendregt, Hendrik Pieter. The Lambda Calculus: Its Syntax and Semantics.North-Holland (Amsterdam, 1981). ISBN 0-444-85490-8. Studies in Logic and the Foundations of Mathematics, vol. 103.

Chaitin, Gregory J. The Limits of Mathematics:A Course on Information Theory and the Limits of Formal Reasoning. Springer-Verglag Singapore (Singapore: 1998). ISBN 981-3083-59-X.

Chaitin, Gregory J.The Unknowable.Springer-Verlag Singapore(Singapore: 1999). ISBN 981-4021-72-5 (hardcover)

Church, Alonzo. An Unsolvable Problem of Elementary Number Theory. American Journal of Mathematics 58 (1936), 345-363. Reprinted in pp. 88-107 of [Davis1965]

Davis, Martin (ed.).The Undecidable: Basic Papers on Undecidable Propositions,Unsolvable Problems and Computable Functions. Raven Press (New York: 1965). ISBN 0-911216-01-4.

Davis, Martin. Computability and Unsolvability. Dover (New York: 1958, 1973, 1982). ISBN 0-486-61471-9 pbk.

Garey, Michael R., Johnson, David S. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman (New York: 1979). ISBN 0-7167-1045-5 pbk.

John E.,Motwani, Rajeev.,Ullman, John D. Introduction to Automata Theory, Languages, and Computation. ed.2. Addison-Wesley (Boston, MA: 2001). ISBN 0-201-44124-1.

Garey, Michael R., Johnson, David S. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman (New York: 1979). ISBN 0-7167-1045-5 pbk. See [Garey1979]

Lewis, Harry R., Papadimitriou, Christos H. Elements of the Theory of Computation. Prentice-Hall (Englewood Cliffs, NJ: 1981). ISBN 0-13-273417-6.

Lifschitz, Vladimir (ed.) Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy. Academic Press (San Diego: 1991). ISBN 0-12-450010-2.

Hopcroft, John E., Motwani, Rajeev., Ullman, John D. Introduction to Automata Theory, Languages, and Computation. ed.2. Addison-Wesley (Boston, MA: 2001). ISBN 0-201-44124-1. See [Hopcroft2001]

Lewis, Harry R., Papadimitriou, Christos H. Elements of the Theory of Computation. Prentice-Hall (Englewood Cliffs, NJ: 1981). ISBN 0-13-273417-6. See [Lewis1981]

Révész, Gy.rgy E. Lambda-Calculus, Combinators and Functional Programming. Cambridge University Press (Cambridge, 1988). ISBN 0-521-34589-8.

Sipser,Michael. Introduction to the Theory of Computation. PWS Publishing (Boston, MA: 1997). ISBN 0-534-94728-X.

Hopcroft, John E., Motwani, Rajeev., Ullman, John D. Introduction to Automata Theory, Languages, and Computation. ed.2. Addison-Wesley (Boston, MA: 2001). ISBN 0-201-44124-1. See [Hopcroft2001]